Exhibit 51

What is a prop bet? (US)

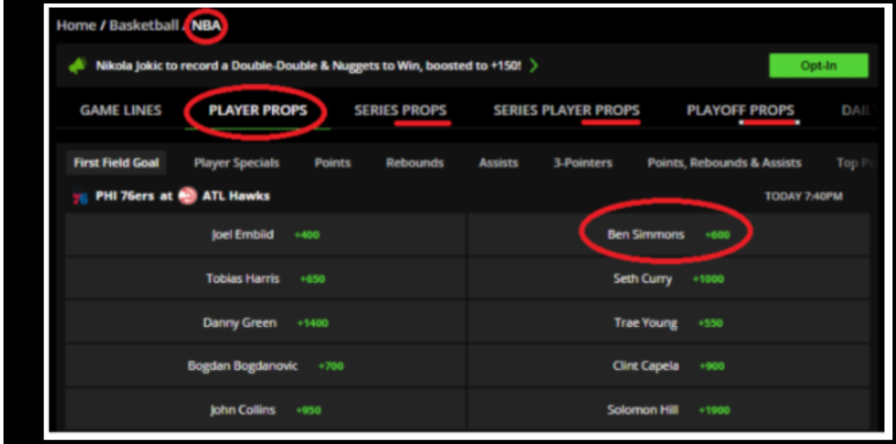
A prop bet or proposition bet is a wager on an individual player or specific event in a game/match that is not directly connected to the outcome of a game. This type of wagering allows you to place wagers on outcomes within the game rather than on the game as a whole e.g., points spread, moneyline, and/or totals. Learn more about how player props are settled.

Prop bet examples:

- "Will LeBron James score over or under 32 points?"
- "Will Julio Jones score a touchdown in this game?"

You can find player props if you visit any individual league page and navigate to the Player Props section.

In the below example, we have navigated to the NBA league page and selected the 'Player Props' category. From here we have selected a player prop wager on Ben Simmons to score the first field goal of the game at +600. A $100 bet would net you $600 in profit for this example.



PHI 76ers at ATL Hawks TODAY 7:40PM

Joel Embiid	+400	Ben Simmons	+600
Tobias Harris	+650	Seth Curry	+1000
Danny Green	+1400	Trae Young	+550
Bogdan Bogdanovic	+700	Clint Capela	+900
John Collins	+950	Solomon Hill	+1900

Was this article helpful? Yes No

JUMP TO SECTION

- Getting Started with Sportsbook
- How to Play - Sportsbook
- Sportsbook Basics
- My Sportsbook Account
- DK Horse

Additional Ways to Contact Us

Mail:

How can we help?

A prop bet or proposition bet is a wager on an individual player or specific event in a game/match that is not directly connected to the outcome of a game. This type of wagering allows you to place wagers on outcomes within the game rather than on the game as a whole e.g., points spread, moneyline, and/or totals. Learn more about how player props are settled.

My Sportsbook Account

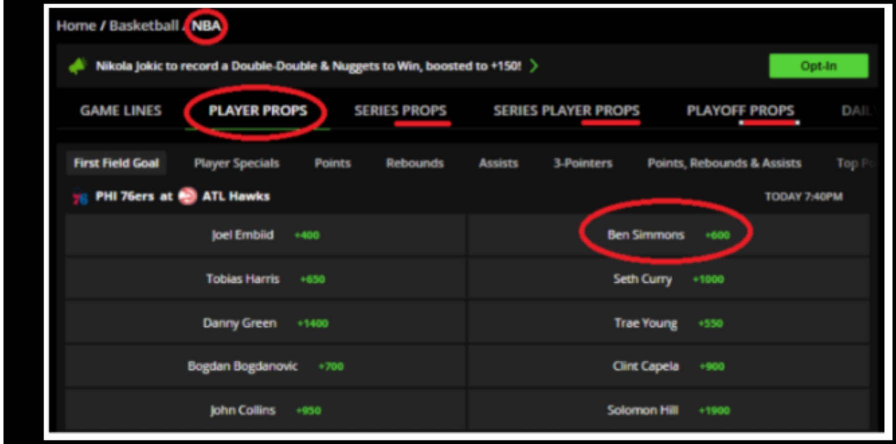
DK Horse

Prop bet examples:

- "Will LeBron James score over or under 32 points?"
- "Will Julio Jones score a touchdown in this game?"

You can find player props if you visit any individual league page and navigate to the Player Props section.

In the below example, we have navigated to the NBA league page and selected the 'Player Props' category. From here we have selected a player prop wager on Ben Simmons to score the first field goal of the game at +600. A $100 bet would net you $600 in profit for this example.



Home / Basketball / NBA

Nikola Jokic to record a Double-Double & Nuggets to Win, boosted to +150! Opt-In

GAME LINES PLAYER PROPS SERIES PROPS SERIES PLAYER PROPS PLAYOFF PROPS DAI

First Field Goal Player Specials Points Rebounds Assists 3-Pointers Points, Rebounds & Assists Top P

PHI 76ers at ATL Hawks TODAY 7:40PM

Joel Embiid +400	Ben Simmons +600
Tobias Harris +650	Seth Curry +1000
Danny Green +1400	Trae Young +550
Bogdan Bogdanovic +700	Clint Capela +900
John Collins +950	Solomon Hill +1900

Was this article helpful? Yes No

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

How can we help?

Exhibit 52

WinView Raises $12 Million in Series B Funding to Expand Platform Where Fans Predict Sports Live While Watching TV

Graham Holdings, Discovery Communications, Ted and Zach Leonsis of Monumental Sports & Entertainment and LionTree Partners join round to help fill a sports culture void by converting fans from spectators into play-along-live participants

NEWS PROVIDED BY
WinView, Inc. →
May 15, 2017, 06:30 ET

SAN FRANCISCO, May 15, 2017 /PRNewswire/ -- WinView, Inc. (**www.winviewgames.com**), the nation's #1 second-screen live TV sports prediction platform, announced a $12 million Series B funding round with participation from a first-class investor group, including Graham Holdings Company, Discovery Communications, Ted Leonsis' Monumental Sports & Entertainment and LionTree Partners. The funding will help WinView further develop and market its skill-based mobile app across a range of professional sports, including baseball, basketball and football (MLB, NBA and NFL), by converting fans and TV viewers from spectators into play-along-live participants.

WinView Raises $12 Million in Series B Funding to Expand Platform Where Fans Predict Sports Live While Watching TV

The WinView Games app brings together the worlds of live TV sports, mobile apps, games of skill and interactive advertising so that sports fans and fantasy players can enjoy sports on TV in a new way by having more fun and interacting with their favorite teams, and with each other. Available on iOS and Android, WinView Games allows users to play along in real time with live televised sports and compete to accurately predict the plays on the field in order to win cash prizes. WinView Games is an ad-supported app that is 100% free to play. The games covered by WinView are produced in a live studio by a team of producers based on what is happening in real time.

Investors in the Series B round include:

- Graham Holdings Company (NYSE: GHC): A diversified holding company whose principal operations include Graham Media Group, which owns seven television broadcast stations in major markets and Kaplan, a global provider of educational services. The Company also owns Slate, an on-line magazine; Panoply, a premium podcast network; SocialCode, a leading social marketing solutions company; Graham Healthcare Group, and several manufacturing businesses.
- Discovery Communications (NASDAQ: DISCA): A leading global media and entertainment company reaching 3 billion cumulative viewers with a portfolio of premium nonfiction, lifestyle, sports and kids content brands. WinView looks forward to pursuing opportunities to deepen its relationship with Discovery and its content brands, including Eurosport the leading provider of locally relevant, premium sports across Europe.
- Ted Leonsis, Founder of Monumental Sports & Entertainment, and Zach Leonsis: Own and operate five professional sports teams (Washington Capitals, Washington Wizards, Washington Mystics, Washington Valor and Baltimore Brigade) and the Verizon Center in Washington, D.C. Ted Leonsis also serves on the board of governors for the NBA and NHL, and he was previously president of AOL (1993 to 2006).
- LionTree Partners: The merchant banking arm of LionTree LLC, a differentiated global investment and merchant banking firm founded in 2012, with a focus on media, technology and telecommunications.

The funding round was driven by cable television and next-gen TV pioneer Tom Rogers, who is Executive Chairman of WinView, Inc. Previously Mr. Rogers was Chairman, President and CEO of TiVo, President of NBC Cable and Executive Vice President of NBC and Chairman and CEO of PRIMEDIA, Inc.

"This first-class group of investors is a major validation of our live TV sports prediction platform and its upside potential to engage TV viewers in a new way to enjoy televised sports," said Tom Rogers, Executive Chairman of WinView, Inc. "The WinView Games app fills a sports culture void by converting fans and TV viewers from spectators into play-along-live participants. Following last fall's successful start, these investments will enable us to take a significant step forward toward reaching our mobile, social and gaming business goals while capitalizing on fan excitement across virtually all professional sports including basketball, baseball and football. We also will strive to be part of the in-stadium experience, enabling fans to predict plays from their seats."

"The WinView Games app is changing the way fans watch live TV sports," said David Lockton, founder and CEO of WinView, Inc. "Our two-screen technology now backed by 40 patents enables sports fans to get closer to the action and create social connections while testing their knowledge of the sport through in-game predictions. With a skill-based multi-sport platform that offers immediate gratification, we are fueling competition among fans while also driving greater tune-in to live televised sports."

Added Lockton, "Predicting what will happen next is core to the sports viewing experience, as fans are constantly thinking ahead during the game and trying to anticipate the action on the field. As the changing landscape of sports and technology continues to evolve with millennials defining the dual-screen age, the 85% of sports viewers who watch sports with their mobile device in hand, can now get into a free game and win money while testing their knowledge."

Timothy O'Shaughnessy, President and CEO of Graham Holdings Company, said: "WinView's platform is a fun, exciting and dynamic new way for sports fans to engage during the most popular live programming on TV today. WinView is well positioned to establish its product as the new social interactive paradigm where fans and friends are competing to predict what's going to happen from the first to the very play of the game."

David Zaslav, President and CEO of Discovery Communications, said: "WinView's products complement Discovery's goal to reach and engage our superfans around the world and across all screens. The company has strong expertise in mobile and sports; a business and engagement model poised to help drive consumption of live content; and a top-notch leadership team. We are pleased to participate in this round of funding to support the growth and development of their platform."

Ted Leonsis, founder and CEO of Monumental Sports & Entertainment, said: "As a team owner, any avenue that creates a greater relationship between the fan and their favorite team and players, as well as greater fan excitement and intimacy, is what local sports franchises are all about. It's my belief that WinView locks fans more deeply into the game experience and rounds out their desire to interact with other fans and prove their knowledge, predictive power and understanding of the sport."

Zach Leonsis, Vice President and General Manager of Monumental Sports Network, said: "WinView really capitalizes on what so many fans are already doing: scrolling around on their mobile devices while they watch games on the big screen. It makes fans a part of the action and it drives huge user engagement. That's a big piece of what's so powerful about real-time sports gaming – it gives fans the chance to become players, and gives us all a shared experience that builds on the intense passion of being a sports fan."

Aryeh B. Bourkoff, founder and CEO of LionTree LLC, said: "In our role as investor and advisor, we recognize that delivering content, particularly to younger audiences, in new, interactive and exciting ways is critical to success. WinView's platform provides users with a unique and innovative experience that deepens their engagement with content, and one another, in real time. Given the success of play-along mobile sports and gaming in Europe, we are confident that WinView has a significant opportunity to advance this trend in the U.S. and we are excited to be a part of driving this business forward."

Launched in 2014, WinView Games previously raised $8 million and represents the next evolution of two-screen TV entertainment in the U.S, as in-game sports predictions already has caught on and become more popular than daily fantasy sports in Europe.

About WinView Inc.

WinView, Inc. is a Silicon Valley and New York based company that is focused on free mobile two-screen synchronized televised sports games for prizes in the U.S. The Company plans to leverage its extensive experience in pioneering real-time interactive television games played on the mobile second screen, its foundational patents and unique business model. The WinView app is an end-to-end two-screen TV synchronization platform for both television programming and commercials. The free skill-based WinView Games app uniquely enhances TV viewing enjoyment and rewards sports fans with prizes as they answer in-game questions while competing with friends in real time during live televised sports. These free games of skill are legal in every state. For more information, please visit **www.winviewgames.com**.

About WinView Inc.'s Intellectual Property

WinView's portfolio of 40 foundational patents covers the synchronization of the second-screen with TV broadcasts and commercials and the optimum methods of monetizing sports-based games of skill, among other types of programming content. The WinView platform provides sponsors and advertisers with exclusive access to all the two-screen game participants through sponsorship of the game, in-game videos and incentivized interactive commercials in sports telecasts. The Company plans to expand its service in partnership with the leagues, TV and cellular industries, leveraging its intellectual property to establish a two-screen synchronization standard across the entire TV broadcast and cellular networks, which are industries that rely upon standards to function efficiently at scale.

Contact Info

Media
Business: Whit Clay, Sloane & Company
T: 212-446-1864
M: 917-601-6012
wclay@sloanepr.com

Consumer/Sports: Andrew Bard, Burson-Marsteller
T: 212-614-5274
M: 646-525-5517
Andrew.Bard@bm.com

Corporate: Anthony Giombetti, WinView
T: 818-821-7530
anthony@winview.tv

SOURCE WinView, Inc.

Related Links

http://www.winviewgames.com

Exhibit 53

Global Baseball v2

Statistics Summary

Documentation

OpenAPI Specification

Change Log

Coverage Matrix

Frequently Asked Questions

Quick Access	
Postman Collection	**XSD Schema**
▶ Run in Postman	Download

Global Baseball API Overview

The Global Baseball API provides real-time, inning-by-inning scoring (when available) and a database of supplementary statistics. Data is collected via Sportradar on-venue scouts and in-house operators.

League coverage includes: MLB, NPB, KBO, World Baseball Classic, and more. Select the **Global Baseball** package in our Coverage Matrix for competitions and data offered.

The API is consistent in structure, format and behavior with the other General Sport APIs. Primary feeds will return seasons, competitions, team and player data, and real-time scores.

Additional feeds provide a host of complementary stats, including:

- Standings
- Player profiles
- Team profiles
- Historical results
- Match win probabilities

An extended Probabilities package is also offered. This add-on includes in-game probability updates and season outrights.

API	API Version

Global Baseball	v2

Note: Authentication is required for all API calls.

Global Baseball API Map

To best utilize the Global Baseball API, you will need several parameters to create your API calls. The map below illustrates how you can obtain the parameters you need.

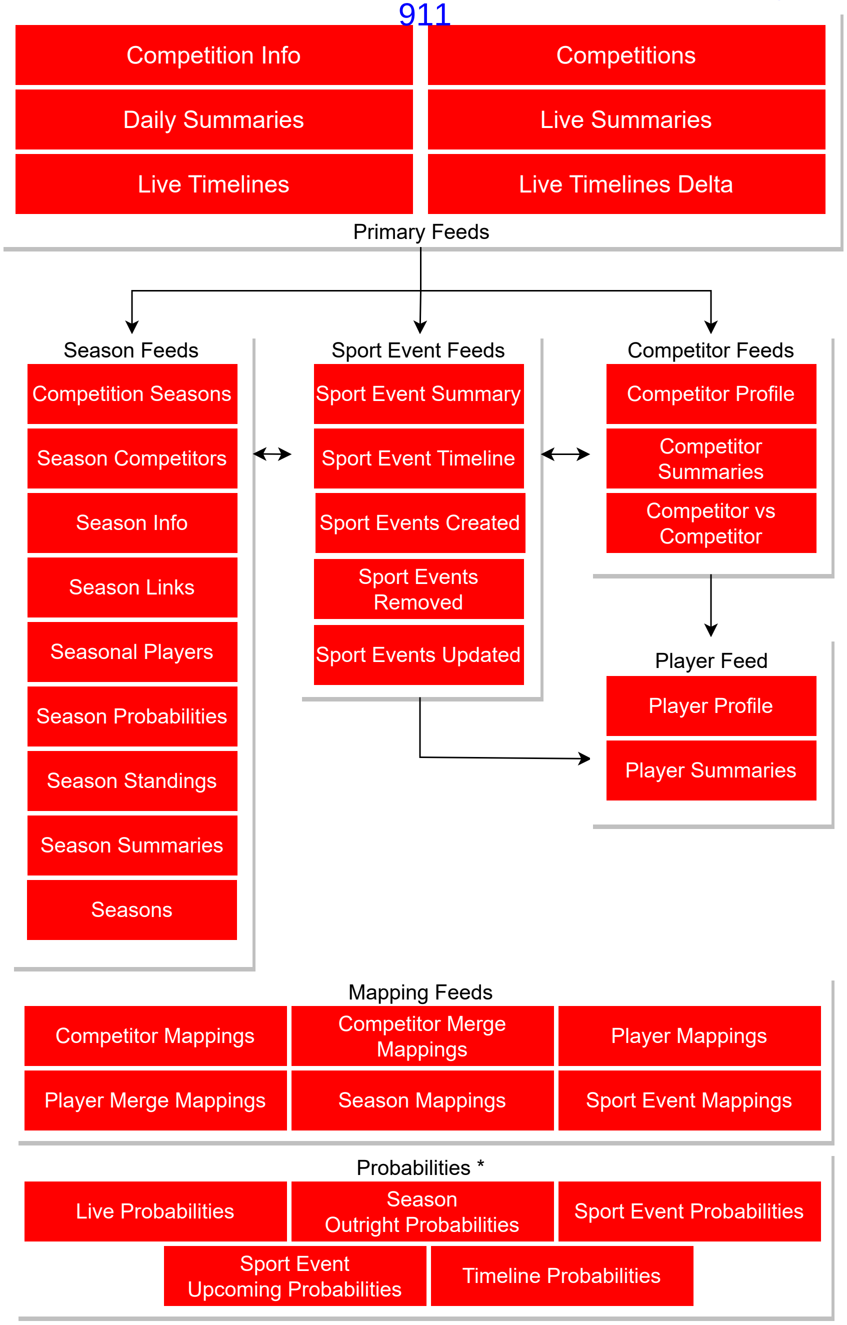
Example:

To find the probability of a home team win for a given game:

1. Call the Daily Schedule for the day the sport event takes place and find the Season Id for the chosen sport event
2. Make note of the Sport Event Id for the given sport event
3. Call the Season Probabilities using the Season Id
4. Find Sport Event Id within the results locate the home team and the outcome probability

The probability of a home team win is displayed.

The primary feeds require only a date or human-readable parameters to call the endpoints. Those feeds provide Sport Event Ids, Competitor Ids, or Season Ids which can be used to generate the match, team, and tournament feeds.



- - Available only with the probabilities plan within the Global Baseball v2 package. See FAQ for details.

Competition Info

Provides the name, id, and parent id for a given competition.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name

Competition Info Data Points:	Gender Id	Name	Parent Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitions/sr:competition:657/info.
xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /competitions/ `{competition_id}` /info. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competition Seasons

Provides historical season information for a given competition. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitions/sr:competition:109/seaso
ns.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/competitions/`{competition_id}`/seasons.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitions

Provides a list of all available baseball competitions.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Gender Id	Name	Parent Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitions.xml?api_key={your_api_ke
y}"
```

The above command returns xml structured like this.

Access the Competitions feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/competitions.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Mappings

Provides team id mapping between the Global Baseball API and the MLB API.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitors/mappings.xml?api_key={you
r_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/competitors/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.

`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/competitors/mappings.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Competitor Merge Mappings

Provides the valid Sportradar Id in cases when two competitors have been merged into one. Mapping entries will remain in the feed for 7 days.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	Name	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/baseball/trial/v2/en/competitors/merge_mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/competitors/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Profile

Provides top-level information for a given team, including the full team roster, manager, home venue, and team colors.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name Jersey Base Color Jersey Horizontal Stripes	Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Sleeve Detail Jersey Split Jersey Squares Jersey Stripes	Jersey Type Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name Manager Nationality
Player Info Data Points:	Country Code Date of Birth Gender Height	Id Jersey Number Nationality	Name Position Weight
Venue Info Data Points:	Capacity City Country Code	Country Name Id	Map Coordinates Name

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitors/sr:competitor:3637/profil
```

```
e.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Summaries

Provides previous and upcoming game information for a given team, including statistics for past games and scheduling info for upcoming games.

TTL / Cache:		**300 seconds**	
Update Frequency:		As Necessary	
Content Type:		XML or JSON	
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name

Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitors/sr:competitor:419265/summ
aries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor vs Competitor

Provides previous and upcoming matches between two teams, including scoring information.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/competitors/sr:competitor:419265/vers
us/sr:competitor:419263/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor vs Competitor feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/competitors/`{competitor_id}`/versus/`{competitor2_id}`/summaries.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`competitor2_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Daily Summaries

Provides game information for a given day, including scoring information.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout

		Normaltime Score Sport Event Status Away Score	Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/schedules/2019-04-24/summaries.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Daily Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /schedules/ `{year}` - `{month}` - `{day}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`year`	Year in 4 digit format (YYYY).
`month`	Month in 2 digit format (MM).
`day`	Day in 2 digit format (DD).
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/schedules/2019-04-24/summaries.xml?api_key={y
our_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 200.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description

`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Live Summaries

Provides game information for all currently live games, including team scoring. This feed updates in real time as games are played.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/schedules/live/summaries.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Live Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Live Timelines

Provides a play-by-play event timeline for currently live games.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Status	Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score	Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time
Timeline Info Data Points:	Away Score Break Name Competitor	Home Score Id Period	Period Name Time Type

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/schedules/live/timelines.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the XML Schema Definition (.XSD) for the Match Timeline use the following URL.

https://schemas.sportradar.com/bsa/v1/endpoints/baseball/match_timeline.xsd

Return to top

Live Timelines Delta

Provides a 10 second live delta of game information, including scoring and a play-by-play event timeline.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Status	Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score	Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time
Timeline Info Data Points:	Away Score Break Name Competitor	Home Score Id Period	Period Name Time Type

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/schedules/live/timelines_delta.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines Delta feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines_delta.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Mappings

Provides player id mapping between the Global Baseball API and the MLB API.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 player mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/players/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/players/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/players/mappings.xml?api_key={your_api_key}&s
tart=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Merge Mappings

Provides valid ids for players who have had their profiles merged. While Sportradar always strives to provide one unique player id, it is a possibility for two ids to be created. This feed provides the correct id once profiles have been merged.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mapping Info Data Points:	Merged Id	Name	Retained Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/players/merge_mappings.xml?api_key={y
our_api_key}"
```

The above command returns xml structured like this.

Access the Player Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/players/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/players/merge_mappings.xml?api_key={your_api_
key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Profile

Provides player biographical information, including current team.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Player Info Data Points:	Active Country Code Date Of Birth End Date Gender	Height Id Jersey Number Nationality Name	Position Role Type / Position Start Date Weight

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/players/sr:player:841933/profile.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/players/`{player_id}`/profile.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`player_id`	ID of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Summaries

Provides game info and results for the past 10 games in which a given player participated.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/players/sr:player:841933/summaries.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es)

	Click here for a tabular list of available languages per competition.
`player_id`	ID of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Competitors

Provides a list of teams participating for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Id	Name	Short Name

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/competitors.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Competitors feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /competitors. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Info

Provides detailed information for a given season, including participating teams and league structure.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Group - Group Name Group Id Group Max Rounds Group Name Info Competition Status	Info Venue Reduced Capacity Info Venue Reduced Capacity Max Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/info.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /info. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Links

Provides information about linked cup rounds for a given season.

Use this feed to compile full advancement brackets for relevant seasons/tournaments. Links between all matches and rounds are available when competitors (TBD vs. TBD) are not yet known.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Cup Round Id Cup Round Name Cup Round Order Cup Round State Cup Round Type Cup Round Winner Id Group - Group Name	Group Id Stage End Date Stage Phase	Stage Start Date Stage Type Stage Year
Match Info Data Points:	Sport Event Id	Start Time	Start Time Confirmed

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/stages_groups
_cup_rounds.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Links feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/stages_groups_cup_rounds.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Mappings

Provides season id mapping between the Global Baseball API and the MLB API.

TTL / Cache:	300 seconds	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/mappings.xml?api_key={your_ap
i_key}"
```

The above command returns xml structured like this.

Access the Season Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/seasons/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/seasons/mappings.xml?api_key={your_api_key}&s
tart=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Season Players

Provides names and ids for all participating players for a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 800 players.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Player Info Data Points:	Display First Name Display Last Name	First Name Id	Last Name Name

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/players.xml?a
pi_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Players feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/players.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es)

	Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/players.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 800.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 800. Example: limit=75

Return to top

Season Probabilities

Provides 2-way win probabilities (home team win, away team win) for all games for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id	Round Number Season Competition Id Season End Date Season Id	Stage End Date Stage Order Stage Phase Stage Start Date

	Coverage Live Group - Group Name Group Id Group Name	Season Name Season Start Date Season Year	Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability	Market Name	Outcome Name
Match Info Data Points:	Competitor Qualifier Sport Event Id	Sport Event Replaced By Sport Event Resumed Time	Sport Event Start Time Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/probabilitie
s.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/baseball/trial/v2/en/seasons/sr:season:61503/probabilities.xml?ap
```

```
i_key={your_api_key}&start=200'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Season Standings

Provides detailed standings info for a given season.

TTL / Cache:	**10 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Group - Group Name Group Live Group Id Group Name	Stage End Date Stage Order Stage Phase	Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Standings Info Data Points:	Current Outcome Draw Games Behind Group - Group Name Group Id Group Live	Group Name Losses Played Runs Against Runs Difference	Runs For Rank Season Standings Tie Break Rule Season Standings Type Winning Percentage Wins

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/standings.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Standings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /standings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Summaries

Provides schedule information and scoring for all matches from a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name

Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/summaries.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/seasons/sr:season:61503/summaries.xml?api_key
={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Seasons

Provides a list of historical season information for all competitions. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/seasons.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/seasons.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.

`your_api_key`	Your API key.

Return to top

Sport Event Mappings

Provides game id mapping between the Global Baseball API and the MLB API.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 sport event mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/sport_events/mappings.xml?api_key={yo
ur_api_key}"
```

The above command returns xml structured like this.

Access the Season Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/sport_events/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Event Summary

Provides real-time scoring for a given game. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/sport_events/sr:sport_event:17433095/
summary.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Summary feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{match_id}` /summary. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`match_id`	ID of a given match.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Timeline

Provides real-time scoring and a play-by-play event timeline for a given game. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name

Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Timeline Info Data Points:	Away Score Break Name Competitor	Home Score Id Period	Period Name Time Type

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/sport_events/sr:sport_event:17433095/
timeline.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /timeline. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.

`sport_event_id`	ID of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Events Created

Provides ids for sport events that have been created in the last 24 hours.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport Event Info Data Points:	Active Season Flag	Id	Created At

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/sport_events/created.xml?api_key={you
r_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Created feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/sport_events/created.`{format}`?api_key=`{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/sport_events/created.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Removed

Provides ids for sport events that have been removed from the API due to an entry error. Ids will remain in the response for 2 weeks.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Removed Id

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/sport_events/removed.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Removed feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/sport_events/removed.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/sport_events/removed.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Updated

Provides ids for sport events that have been updated in the last 24 hours.

TTL / Cache:	**60 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Match Info Data Points:	Id	Updated At

```
curl -X GET "https://api.sportradar.us/baseball/trial/v2/en/sport_events/updated.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Updated feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball/`{access_level}`/`{version}`/`{language_code}`/sport_events/updated.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/baseball/trial/v2/en/sport_events/updated.xml?api_key={your_api_ke
y}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Probabilities Feeds

This collection of probability feeds is available for licensing as an add-on feature. They are an extension of the Season Probabilities feed in the main package which provides pre-match probabilities for the sport event winner market.

For media use only: Use of this API is prohibited for betting clients.

The main features of the Probabilities extension (where coverage applies) are:

- Live Probabilities that update throughout game
- Season Outright Probabilities for the Tournament Winner market
- Live Probabilities Coverage indicator for the next 24 hours

For more information or to request a Trial of this feature, please contact a Sportradar Sales representative.

The Open API specification and Syntax for using our Probabilities feeds, including examples of the payloads can be found below.

OpenAPI Specification

Live Probabilities

Provides top-level information for live games. If probabilities are available for a game, pre-match and live probabilities will be displayed.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score	Market Home Score Market Last Updated Market Live	Market Name Market Removed Outcome Name
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball-probabilities/trial/v2/en/schedules/live/probabil
ities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball-probabilities/ `{access_level}` / `{version}` / `{language_code}` /schedules/live/probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Outright Probabilities

Provides a list of outright probabilities for each team from a given season.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON

Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Probability Info Data Points:	Win Probability	Market Name	Outcome Name

```
curl -X GET "https://api.sportradar.us/baseball-probabilities/trial/v2/en/seasons/sr:season:7950
7/outright_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Outright Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball-probabilities/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/outright_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Probabilities

Provides pre-match and live probabilities for a given match.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id	Round Number Season Competition Id Season End Date Season Id	Stage End Date Stage Order Stage Phase Stage Start Date

	Coverage Live Group - Group Name Group Id Group Name	Season Name Season Start Date Season Year	Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score	Market Home Score Market Last Updated Market Live	Market Name Market Removed Outcome Name
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball-probabilities/trial/v2/en/sport_events/sr:sport_e
vent:24142067/sport_event_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.

`your_api_key`	Your API key.

[Return to top](#)

Sport Event Upcoming Probabilities

Provides a list of IDs for upcoming sport events in the next 24 hours.

TTL / Cache:	**1 second**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Upcoming Id

```
curl -X GET "https://api.sportradar.us/baseball-probabilities/trial/v2/en/sport_events/upcoming_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like [this](#).

Access the Sport Event Upcoming Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/upcoming_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click [here](#) for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

[Return to top](#)

Timeline Probabilities

Provides a timeline of pre-match and live probability changes for a given game.

TTL / Cache:	**1 second**

Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score	Market Home Score Market Last Updated Market Live	Market Name Market Removed Outcome Name
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id	Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Inning Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/baseball-probabilities/trial/v2/en/sport_events/sr:sport_event:24142067/timeline_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Timeline Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/baseball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/timeline_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), English (en), Japanese (ja), Korean (ko), Spanish (es) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Frequently Asked Questions

Q: What leagues do you cover, and at what level?

Coverage information can be found via our Coverage Matrix.

Q: What format are date fields presented in?

A: When we present date only values we present these in the ISO 8601 standard format.
ex: 2013-04-03
We use these for attributes that have date and no time (such as birthdate). For more information:
https://en.wikipedia.org/wiki/ISO_8601

Q: What format are the date/time fields presented in?

A: All of our Date/Time attributes are in UTC, presented in the ISO 8601 standard format.
ex: 2013-04-03T18:15:00+00:00
For more information: https://en.wikipedia.org/wiki/ISO_8601

Q: How can I find the values for various enum data points within the API?

A: Many enum values are listed in the FAQ below. For the most up-to-date values, please see the Schema section of the OpenAPI specification here: https://api.sportradar.com/baseball/trial/v2/openapi/swagger/index.html

Q: What are the valid sport_event_status - status values?

A: Here are the valid sport_event_status values and their definitions:

- not_started – The match is scheduled to be played
- match_about_to_start – The match is about to begin
- started – The match has started
- live – The match is currently in progress
- postponed – The match has been postponed to a future date
- suspended - The match began, but has been suspended to a later time
- delayed – The match has been temporarily delayed and will be continued

- cancelled – The match has been cancelled and will not be played
- interrupted - The match began, but coverage has stopped for a short time. Note that match scores may not be updated during this period, the last recorded match score will be displayed instead
- ended – The match is over
- closed – The match results have been confirmed

Q: What are the valid sport_event_status – match_status values?

A: Here are the valid match_status values:

- not_started
- start_delayed
- aet
- interrupted
- ended
- postponed
- started
- abandoned
- cancelled
- overtime
- 1st_inning
- 2nd_inning
- 3rd_inning
- 4th_inning
- 5th_inning
- 6th_inning
- 7th_inning
- 8th_inning
- 9th_inning
- pause

Q: What are the possible event types logged?

A: Here are the valid event type:

- match_started
- period_start
- score_change
- period_score
- break_start
- match_ended

Q: How do I access past seasons results and stats?

A: Use the Competition Seasons endpoint to locate the season_id for the season you want to access. Use that season_id to interrogate the various seasons endpoints.

Q: What are the valid outcomes for probabilities?

A: Here are the valid outcome probabilities:

- home_team_winner
- away_team_winner

Q: How do I find out the coverage for a particular match?

A: Find the node called "coverage" in any of the Summary, Lineups, or Timeline feeds. The attribute "coverage - live" reports if Sportradar has live coverage of the match or not.

Q: Why do different groups have coverage information for a competition?

A: The notion of “groups” is versatile and is used to distinguish between playoffs, and our competition structures and is therefore necessary to describe coverage at a group level for consistency. Generally, however, within a competition in Baseball, there will be no difference between competition coverage.

Q: What are markets and what are the different markets?

A: Markets is something you can bet on that we provide probabilities for. Over time we intend to provide more and more markets in the API. Currently the only market we provide is 2-way (will the home team win? Or the away team?).

Q: How are group IDs delivered in the stage array with the various types?

A: With the type of "league" they will have a sr:league prefix. With the type of "cup" they will have a sr:cup prefix.

Q: How are “live” feeds handled in the API?

A: Sport Events appear in the feed 10 minutes before the scheduled start time and are removed 10 minutes after the Sport Event is ended.

Q: What is the probabilities package?

A: The probabilities package is an add-on set of feeds that are an extension of the Season Probabilities feed in the main package (which already provides pre-match probabilities for the sport event winner market). The main features of the Probabilities extension are: Live Probabilities that update throughout game, Season Outright Probabilities for the Tournament Winner market, and Live Probabilities Coverage indicator for the next 24 hours.

Q: Is there Live Probability coverage for every game you cover in the Global Baseball API?

A: Live Probability coverage depends on a number of factors and we can’t guarantee that every game will be covered for any given league, however, Sportradar aims to cover close to 100% of games from the following Tier 1 leagues: Major League Baseball (MLB) – USA, Nippon Professional Baseball (NPB) – Japan, and KBO League – South Korea

Q: Can I utilize the Probabilities extension if I license the Sportradar MLB API?

A: If you have a license for our MLB API you can also integrate Live Probabilities by utilizing the ID Mapping feeds available in Global Baseball v2 to connect games between the two products.

Q: Why can't I find a particular match in the Daily Summaries, Season Summaries or Sport Events Updated feeds?

A: These endpoints support pagination and return 200 entries by default. To return more matches, an additional query string parameter must be used after your API key. For example, appending *&start=200* will return the next 200 entries per page,

&start=400 will return the next 200, and so on.

Q: How will a sport event behave when it is not covered with live scores?

A: When a sport_event is not covered live, the status and match_status will remain as not_started until results are entered post-match.

Q: How do I locate the TTL (Time to Live)/cache on an API endpoint?

A: The cache (in seconds) can be accessed in the returned header information on each RESTful API call, under `cache-control`. ex. `cache-control: max-age=1, public, s-maxage=1` or `cache-control: public, must-revalidate, max-age=120`

Q: What are the possible values for `cup_round` – `state` in the Season Links feed?

A: Listed below are the values and definitions for `cup_round` - `state`. These can be leveraged to determine the status of a cup round.

- `empty` - A matchup has been created but neither the match details nor the competitors are known.
- `unseeded_fixture` - Match details are known but competitors are unknown.
- `partial_seeded` - One competitor is known.
- `partial_seeded_fixture` - Match details and one competitor are known.
- `seeded` - Both competitors are known.
- `seeded_fixture` - Match details and both competitors are known.
- `unstarted` - Match(es) have been added.
- `on_going` - The first match has started.
- `decided` - The last match has ended.
- `winner` - The winner is known.
- `cancelled` – The matchup has been cancelled.

Q: How are extra innings displayed in the Global Baseball API?

A: In the Summary endpoints, extra inning games have one overtime period for all extra innings.

In the Sport Event Timeline feed, each extra inning will feature `period_name="overtime"`.

Return to top

SPORTRADAR
150 South 5th St. Suite 400
Minneapolis, MN 55402
USA

us-sales@sportradar.com
us-jobs@sportradar.com
Main: 612-361-4100

Sales: 612-361-4102

Terms and Conditions

SPORTRADAR IS SOCIAL

Facebook Technology Partner

Exhibit 54

Global Basketball v2

Statistics Summary

Documentation

OpenAPI Specification

Change Log

Coverage Matrix

Frequently Asked Questions

Quick Access	
Postman Collection	**XSD Schema**
▶ Run in Postman	Download

Global Basketball API Overview

The Global Basketball API provides real-time scoring and statistics (when available) and an array of supplementary data. Data is collected via Sportradar's on-venue scouts and in-house operators.

Over 200 unique competitions are available in one package. Select the **Global Basketball** package in our Coverage Matrix for competitions and data offered.

The API is consistent in structure, format and behavior with the other General Sport APIs. Primary feeds will return seasons, competitions, team and player data, and real-time scores.

Additional feeds provide a host of complementary stats, including:

- Standings
- Player profiles
- Team profiles
- Historical results
- Game rosters
- Seasonal statistics
- Match win probabilities
- Live event positional data

Real-time customers are also offered two delivery Push Feeds to enhance speed.

An extended Probabilities package is also offered. This add-on includes in-game probability updates and season outrights.

API	API Version
Global Basketball	v2

Note: Authentication is required for all API calls.

Global Basketball API Map

To best utilize the Global Basketball API, you will need several parameters to create your API calls. The map below illustrates how you can obtain the parameters you need.

Example:

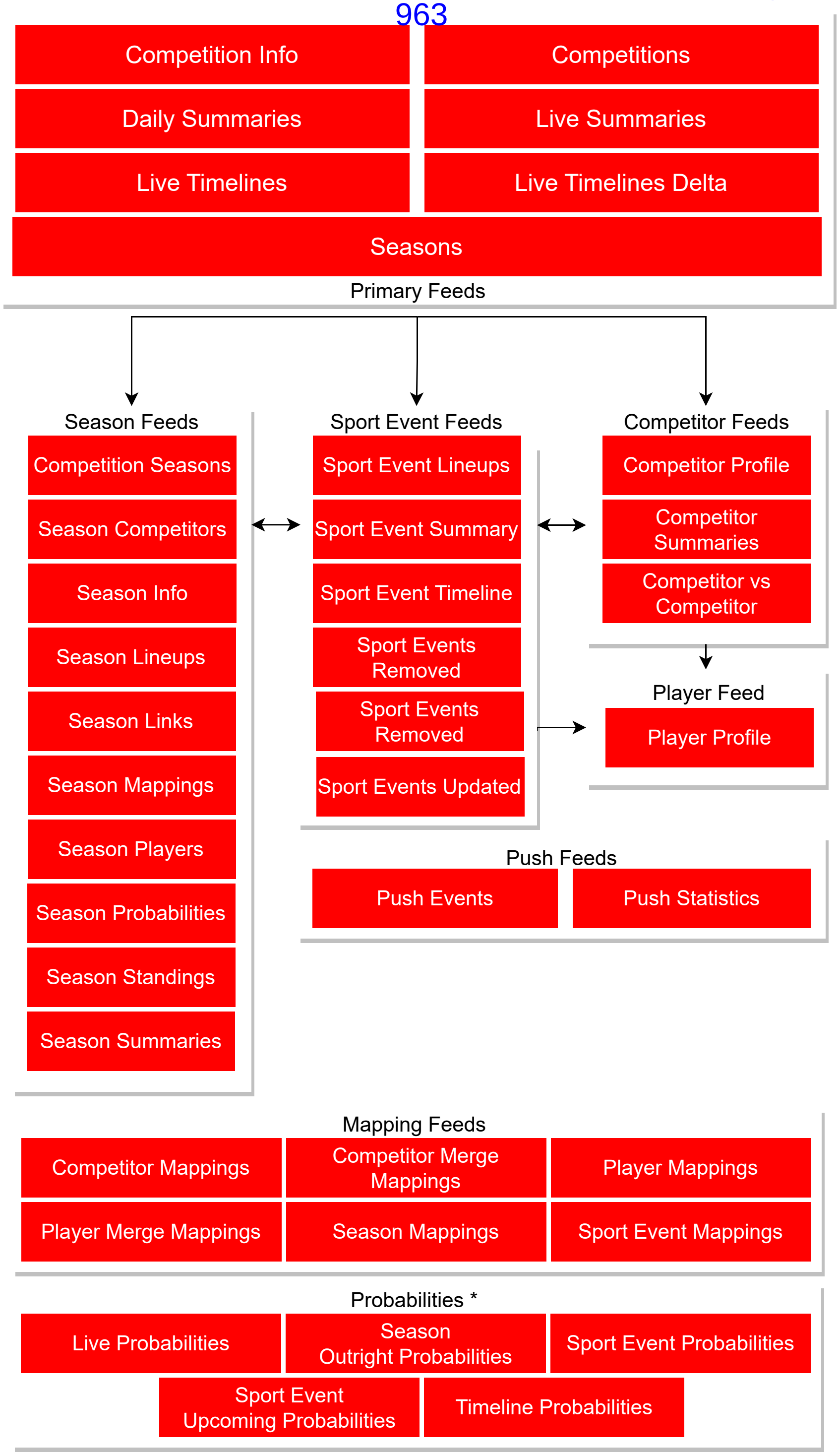
To find the probability of a home team win for a given game:

1. Call the Daily Schedule for the day the sport event takes place and find the Season Id for the chosen sport event
2. Make note of the Sport Event Id for the given sport event
3. Call the Season Probabilities using the Season Id
4. Find Sport Event Id within the results locate the home team and the outcome probability

The probability of a home team win is displayed.

The primary feeds require only a date or human-readable parameters to call the endpoints. Those feeds provide Sport Event Ids, Competitor Ids, or Season Ids which can be used to generate the match, team, and tournament feeds.

- - Available only with the probabilities plan within the Global Basketball v2 package. See FAQ for details.



Competition Info

Provides the name, id, and parent id for a given competition.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name

Competition Info Data Points:	Competition Gender Competition Id	Competition Name	Competition Parent Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitions/sr:competition:138/in
fo.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/competitions/`{competition_id}`/info.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competition Seasons

Provides historical season information for a given competition. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitions/sr:competition:138/se
asons.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /competitions/ `{competition_id}` /seasons. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitions

Provides a list of all available basketball competitions.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Competition Gender Competition Id	Competition Name	Competition Parent Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitions.xml?api_key={your_api
_key}"
```

The above command returns xml structured like this.

Access the Competitions feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/competitions.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.
`parents=true`	Optional query parameter to display only parent competitions. Note: Optional query string parameters must be added after your API key with an ampersand (&).

Return to top

Competitor Mappings

Provides team id mapping between the Global Basketball API and other basketball APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 competitor mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitors/mappings.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Mapping feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /competitors/mappings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/competitors/mappings.xml?api_key={your_api
_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Competitor Merge Mappings

Provides the valid Sportradar Id in cases when two competitors have been merged into one. Mapping entries will remain in the feed for 7 days.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	Name	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitors/merge_mappings.xml?api
_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/competitors/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Profile

Provides top-level information for a given team, including the full team roster, home venue, and team colors.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name

Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Division Competitor Gender Competitor Id Competitor Name Competitor State	Jersey Base Color Jersey Horizontal Stripes Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Split Jersey Squares	Jersey Stripes Jersey Type Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name Manager Nationality
Player Info Data Points:	Country Code Date of Birth Gender Height	Id Jersey Number Nationality	Name Position Weight
Venue Info Data Points:	Capacity City Country Code	Country Name Id	Map Coordinates Name

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitors/sr:competitor:3515/pro
file.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Summaries

Provides previous and upcoming game information for a given competitor, including statistics for past games and scheduling info for upcoming games.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls Free Throw Attempts Successful Free Throw Attempts Made Leader Assists Leader Assists Player Id	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound Rebounds Shots Blocked Steals Team Leads Team Rebounds	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead Timeouts Turnovers Two Point Attempts Successful Two Point Attempts Total

Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitors/sr:competitor:3501/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor vs Competitor

Provides previous and upcoming games between two teams including scoring information, player game statistics and team game statistics.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name

Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls Free Throw Attempts Successful Free Throw Attempts Made Leader Assists Leader Assists Player Id	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound Rebounds Shots Blocked Steals Team Leads Team Rebounds	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead Timeouts Turnovers Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/competitors/sr:competitor:3501/ver
sus/sr:competitor:6676/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor vs Competitor feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /versus/ `{competitor2_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`competitor2_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Daily Summaries

Provides game information for a given day including team scoring, player and team game statistics.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type	Round Number Season Competition Id Season End Date Season Id Season Name	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

	Coverage Property Value Group Id Group Name	Season Start Date Season Year	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls Free Throw Attempts Successful Free Throw Attempts Made Leader Assists Leader Assists Player Id	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound Rebounds Shots Blocked Steals Team Leads Team Rebounds	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead Timeouts Turnovers Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/schedules/2019-01-15/summaries.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Daily Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /schedules/ `{year}` - `{month}` - `{day}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`year`	Year in 4 digit format (YYYY).
`month`	Month in 2 digit format (MM).
`day`	Day in 2 digit format (DD).
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/schedules/2019-01-15/summaries.xml?api_key
={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Live Summaries

Provides game information for all currently live games including team scoring, player and team match statistics. This feed updates in real time as matches are played. Matches appear a few minutes before kick-off and disappear a few minutes after the match reaches "ended" status.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls Free Throw Attempts Successful	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound Rebounds	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead Timeouts

	Free Throw Attempts Made Leader Assists Leader Assists Player Id	Shots Blocked Steals Team Leads Team Rebounds	Turnovers Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/schedules/live/summaries.xml?api_k
ey={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Live Timelines

Provides a play-by-play event timeline for currently live games. Games appear a few minutes before tip-off and disappear a few minutes after the game reaches “ended” status.

TTL / Cache:	**1 second**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Match Info Data Points:	Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Status Sport Event Status Away Normaltime Score	Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status	Sport Event Status Scout Abandoned Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time Time Played Time Remaining
Timeline Info Data Points:	Assist Player Id Assist Player Name Assist Type Away Score Break Name Competitor Home Score	Goal Scorer Id Goal Scorer Name Home Score Id Match Clock Match Time Period	Period Name Time Type X Coordinate Y Coordinate

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/schedules/live/timelines.xml?api_k
ey={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /schedules/live/timelines. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the XML Schema Definition (.XSD) for the Match Timeline use the following URL.

https://schemas.sportradar.com/bsa/v1/endpoints/basketball/match_timeline.xsd

Return to top

Live Timelines Delta

Provides a 10 second live delta of game information, including scoring and a play-by-play event timeline.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Status Sport Event Status Away Normaltime Score	Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status	Sport Event Status Scout Abandoned Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time Time Played Time Remaining
Timeline Info Data Points:	Assist Player Id Assist Player Name Assist Type Away Score Break Name Competitor Home Score	Goal Scorer Id Goal Scorer Name Home Score Id Match Clock Match Time Period	Period Name Time Type X Coordinate Y Coordinate

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/schedules/live/timelines_delta.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines Delta feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /schedules/live/timelines_delta. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Mappings

Provides player id mapping between the Global Basketball API and other basketball APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 player mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/players/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /players/mappings. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi),

	French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/players/mappings.xml?api_key={your_api_ke
y}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Merge Mappings

Provides valid ids for players who have had their profiles merged. While Sportradar always strives to provide one unique player id, it is a possibility for two ids to be created. This feed provides the correct id once profiles have been merged.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mapping Info Data Points:	Merged Id	Name	Retained Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/players/merge_mappings.xml?api_key
={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/players/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/players/merge_mappings.xml?api_key={your_a
pi_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Profile

Provides player information, including team membership info.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Country Code Date Of Birth Gender Height	Id Jersey Number Nationality	Name Position Weight

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/players/sr:player:607296/profile.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`player_id`	ID of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Competitors

Provides a list of teams participating for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation Competitor Id	Competitor Name	Competitor Short Name

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:85212/competitor
s.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Competitors feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/competitors.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Info

Provides detailed information for a given season, including participating teams and league structure.

TTL / Cache:	**300 seconds**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Group Id Group Max Rounds Group Name Info Competition Status	Info Venue Reduced Capacity Info Venue Reduced Capacity Max Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55285/info.xml?a
pi_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /info. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Lineups

Provides game rosters for all games in a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**30 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group - Group Name Group Id	Group Name Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Country Code Date of Birth Height Id	Jersey Number Nationality Name Played	Position Starter Weight
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55467/lineups.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Lineups feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/lineups.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55467/lineups.xml?api_ke
y={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the lineup information with one or more of the following optional query string parameters. By default the limit is set to 200.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Season Links

Provides information about linked cup rounds for a given season.

Use this feed to compile full advancement brackets for relevant seasons/tournaments. Links between all matches and rounds are available when competitors (TBD vs. TBD) are not yet known.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Cup Round Id Cup Round Name Cup Round Order Cup Round State Cup Round Type Cup Round Winner Id Group - Group Name	Group Id Stage End Date Stage Phase	Stage Start Date Stage Type Stage Year
Sport Event Info Data Points:	Sport Event Id	Sport Event Start Time	Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:79153/stages_gro
ups_cup_rounds.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Links feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /stages_groups_cup_rounds. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Links feed, use the following URL.

https://api.sportradar.com/basketball/trial/v2/openapi/swagger/index.html#/seasons/getSeasonStagesGroupsCupRounds

Return to top

Season Mappings

Provides season id mapping between the Global Basketball API and other basketball APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 season mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/mappings.xml?api_key={your
_api_key}"
```

The above command returns xml structured like this.

Access the Season Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/mappings. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/seasons/mappings.xml?api_key={your_api_ke
y}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Season Players

Provides names and ids for all participating players for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Player Info Data Points:	Display First Name Display Last Name	First Name Id	Last Name Name

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:85212/players.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Players feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/players.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Probabilities

Provides 2-way win probabilities (home team win, away team win) for all games for a given season.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability	Market Name	Outcome Name

Match Info Data Points:	Competitor Qualifier Sport Event Id	Sport Event Replaced By Sport Event Start Time	Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:59060/probabilit
ies.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:59060/probabilities.xml?
api_key={your_api_key}&start=200'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description

`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Season Standings

Provides detailed standings info for a given season.

TTL / Cache:	**10 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Group - Group Name Group Live Group Id Group Name	Stage End Date Stage Order Stage Phase	Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Standings Info Data Points:	Change Current Outcome Draw Games Behind Group - Group Name Group Id Group Live Group Name	Last Ten Loss Record Last Ten Win Record Losses Losses Conference Losses Division Played Points Points Against	Points For Rank Season Standings Tie Break Rule Season Standings Type Winning Percentage Winning Percentage Decimal Wins Wins Conference Wins Division

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55467/standings.
xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Standings feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /standings. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55467/standings.xml?api_
key={your_api_key}&round=1'
```

In addition to the URL parameters listed above, you can filter the Standings information with the following optional query string parameter.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

Parameter	Description
`round`	Round number expressed as: {round}. Example: round=1
`live`	Live standings expressed as a boolean value. Example: live=true

To retrieve the OpenAPI Schema Definition for Season Standings, use the following URL.

https://api.sportradar.com/basketball/trial/v2/openapi/swagger/index.html#/seasons/getSeasonStandings

Return to top

Season Summaries

Provides schedule information for all games from a given season including scoring and statistics at the game level.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls Free Throw Attempts Successful Free Throw Attempts Made	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound Rebounds Shots Blocked Steals	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead Timeouts Turnovers

	Leader Assists Leader Assists Player Id	Team Leads Team Rebounds	Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55285/summaries.
xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:55285/summaries.xml?api_
key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Seasonal Competitor Statistics

Provides team seasonal statistics for a given season.

TTL / Cache:	**30 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Sport Id	Sport Name	
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Team Season Statistics Data Points:	Field Goals Made	Minutes	

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons/sr:season:59060/competitor
s/sr:competitor:130098/statistics.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasonal Competitors Statistics feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /competitors/ `{competitor_id}` /statistics. `{format}` ?api_key= `{your_api_key}`

Parameter	Description

`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`competitor_id`	ID of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Seasons

Provides a list of historical season information for all competitions. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season Disabled Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/seasons.xml?api_key={your_api_ke
y}"
```

The above command returns xml structured like this.

Access the Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/seasons.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Lineups

Provides detailed roster information for a given game. Starting players are included if supported by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Country Code Date of Birth Height Id	Jersey Number Nationality Name Played	Type Starter Weight
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score

	Sport Event Id Sport Event Replaced By	Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed	Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/sr:sport_event:151644
19/lineups.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Lineups feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /lineups. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given match.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Mappings

Provides game id mapping between the Global Basketball API and other basketball APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 sport event mappings.

TTL / Cache:	**300 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON

Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/sport_events/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0

`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Event Summary

Provides real-time game-level statistics for a given game. Including scores by quarter and player and team stats. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead	Leader Points Leader Points Player Id Leader Rebounds	Team Turnovers Three Point Attempts Successful

	Defensive Rebounds Fouls Free Throw Attempts Successful Free Throw Attempts Made Leader Assists Leader Assists Player Id	Leader Rebounds Player Id Offensive Rebound Rebounds Shots Blocked Steals Team Leads Team Rebounds	Three Point Attempts Total Time Spent in Lead Timeouts Turnovers Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/sr:sport_event:149835
99/summary.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Summary feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /summary. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given match.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Timeline

Provides real-time game-level statistics and a play-by-play event timeline for a given game. This includes scores by quarter and player and team stats. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code	Competitor Division Competitor Gender Competitor Id	Competitor Name Competitor State
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining
Timeline Info Data Points:	Assist Player Id Assist Player Name Assist Type Away Score Break Name Competitor Home Score	Goal Scorer Id Goal Scorer Name Home Score Id Match Clock Match Time Period	Period Name Time Type X Coordinate Y Coordinate
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead

	Free Throw Attempts Successful Free Throw Attempts Made Leader Assists Leader Assists Player Id	Rebounds Shots Blocked Steals Team Leads Team Rebounds	Timeouts Turnovers Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/sr:sport_event:149722
81/timeline.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /timeline. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Events Created

Provides ids for sport events that have been created in the last 24 hours.

TTL / Cache:	**60 seconds**

Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Active Season Flag	Id	Created At

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/created.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Created feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/created. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/sport_events/created.xml?api_key={your_api
_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Removed

Provides ids for sport events that have been removed from the API due to an entry error. Ids will remain in the response for 2 weeks.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Removed Id

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/removed.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Removed feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/sport_events/removed.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/sport_events/removed.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Updated

Provides ids for sport events that have been updated in the last 24 hours.

TTL / Cache:	**60 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Match Info Data Points:	Id	Updated At

```
curl -X GET "https://api.sportradar.com/basketball/trial/v2/en/sport_events/updated.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Updated feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball/`{access_level}`/`{version}`/`{language_code}`/sport_events/updated.`{format}`?api_key=`{your_api_key}`

Parameter	Description

`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/basketball/trial/v2/en/sport_events/updated.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Push Feeds

To best utilize Push feeds, we have included code samples in Ruby, Java and Python which provide an example of a way you can consume the feeds. Using these samples will output the feeds content to STDOUT.
For Java, we have also provided a Stream Client to assist your integration.

Note: In the provided Java sample, replace "URL GOES HERE" with the desired Push feed URL.

```
package com.sportradar.http.stream.client;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class StreamClientTest {

    private StreamClient client;

    private static String SERVICE_URL = "<url goes="" here="">";

    @Before
    public void setup() {
        client = new StreamClient();
    }

    @After
    public void cleanup() {
        client.terminate();
    }

    @Test
    public void testStream() throws Exception {
        Handler handler = new ConsoleHandler();
        client.stream(SERVICE_URL, handler);
        System.out.println("Connecting....");
        Thread.sleep(1 * 60 * 1000);
        System.out.println("Disconnecting....");
    }

}
```

Some of our APIs include Push feeds that allow you to get updates as soon as they are available. Push API feeds automatically send JSON and XML payloads to you via a push service, and can dramatically reduce the number of calls you need to make to our RESTful API feeds. The structure of the Push feeds are similar to the structure of the corresponding RESTful API feed (i.e. Push Events and Push Statistics). The push service ensures reliable and efficient delivery of the most up to date information.

Our Push services are based on a HTTP publish/subscribe model. When making a call to the Push APIs, you "subscribe" to various data feeds provided by our service; whenever new content is available on one of those feeds, the server pushes that information out to your client. When no new information is available on the feed, a heartbeat message is sent every 5 seconds to keep the connection active. If you want to filter the results of the feeds, there are several optional query string parameters that can be applied to the API call. If left unfiltered, then all data for the feed is displayed (i.e. all events).

For your applications to accept data from our Push feeds, ensure that your application can:

- Can follow a HTTP redirect or use the location provided in the feeds header within one minute of your initial request.
- Can accept HTTP data transfer encoded as chunked.

Our Push service does not provide a "stateful session", there is no memory of what data has been sent previously. If you are disconnected from the Push session, you can use the RESTful API to catch up or recover from the disconnection.

Syntax for using our Push feeds and examples of the JSON and XML payloads can be found below.

Push Events

```
curl -L GET 'api.sportradar.com/basketball/trial/stream/events/subscribe?api_key={your_api_key}'
```

The above command returns json like this.

This endpoint retrieves real-time event updates for all live matches.

https://api.sportradar.com/basketball/`{access_level}`/stream/events/subscribe?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'https://api.sportradar.com/basketball/trial/stream/events/subscribe?api_key={your_ap
i_key}&amp;format=json&amp;sport_event_id=sr:sport_event_id:17582523'
```

In addition to the URL parameters listed above, you can filter the Events information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

URL Parameters	Description
`channel`	Channel type expressed as: {channel_type}. Example: channel=basketball
`competition_id`	Competition Id expressed as: {competition_id}. Example: competition_id=sr:competition:27230
`event_id`	Event type expressed as: {event_type}. Example: event_id=score_change
`format`	Format type expressed as: {format}. Example: format=json
`season_id`	Season id expressed as: {season_id}. Example: season_id=sr:season:59654
`sport_event_id`	Sport event id expressed as: {sport_event_id}. Example: sport_event_id=sr:sport_event:17582523

`sport_id`	Sport id expressed as: {sport_id}. Example: sport_id=sr:sport:22

Return to top

Push Statistics

```
curl -L GET 'api.sportradar.com/basketball/trial/stream/statistics/subscribe?api_key={your_api_key}'
```

The above command returns json like this.

This endpoint retrieves real-time team and player game-level statistics for all live games.

https://api.sportradar.com/basketball/`{access_level}`/stream/statistics/subscribe?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'https://api.sportradar.com/basketball/trial/stream/statistics/subscribe?api_key={your_api_key}&amp;format=json&amp;sport_event_id=sr:sport_event_id:17582523'
```

In addition to the URL parameters listed above, you can filter the Statistics information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

URL Parameters	Description
`channel`	Channel type expressed as: {channel_type}. Example: channel=basketball
`competition_id`	Competition Id expressed as: {competition_id}. Example: competition_id=sr:competition:27230
`event_id`	Event type expressed as: {event_type}. Example: event_id=score_change

`format`	Format type expressed as: {format}. Example: format=json
`season_id`	Season id expressed as: {season_id}. Example: season_id=sr:season:59654
`sport_event_id`	Sport event id expressed as: {sport_event_id}. Example: sport_event_id=sr:sport_event:17582523
`sport_id`	Sport id expressed as: {sport_id}. Example: sport_id=sr:sport:22

Return to top

Probabilities Feeds

This collection of probability feeds is available for licensing as an add-on feature. They are an extension of the Season Probabilities feed in the main package which provides pre-match probabilities for the sport event winner market.

The main features of the Probabilities extension (where coverage applies) are:

- Live Probabilities that update throughout game
- Season Outright Probabilities for the Tournament Winner market
- Live Probabilities Coverage indicator for the next 24 hours

For more information or to request a Trial of this feature, please contact a Sportradar Sales representative.

The Open API specification and Syntax for using our Probabilities feeds, including examples of the payloads can be found below.

OpenAPI Specification

Live Probabilities

Provides top-level information for live games. If probabilities are available for a game, pre-match and live probabilities will be displayed.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live	Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name	Season Start Date Season Year Stage End Date Stage Order Stage Phase

	Coverage Property Type Coverage Property Value Group - Group Name Group Id Group Name	Round Number Season Competition Id Season End Date Season Id Season Name	Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Division Gender	Id Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score	Market Home Score Market Last Updated Market Live	Market Name Market Removed Outcome Name
Match Info Data Points:	Away Score Competitor Qualifier Ground Neutral Home Score Period Number Period Type Sport Event Id Sport Event Replaced By	Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining

```
curl -X GET "https://api.sportradar.com/basketball-probabilities/trial/v2/en/schedules/live/probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball-probabilities/`{access_level}`/`{version}`/`{language_code}`/schedules/live/probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es),

	Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Outright Probabilities

Provides a list of outright probabilities for each team from a given season.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Division	Gender Id Name
Probability Info Data Points:	Win Probability	Market Name	Outcome Name

```
curl -X GET "https://api.sportradar.com/basketball-probabilities/trial/v2/en/seasons/sr:season:79
153/outright_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Outright Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball-probabilities/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /outright_probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.

`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Probabilities

Provides pre-match and live probabilities for a given game.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group - Group Name Group Id Group Name	Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Season Competition Id Season End Date Season Id Season Name	Season Start Date Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Division Gender	Id Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score	Market Home Score Market Last Updated Market Live	Market Name Market Removed Outcome Name
Match Info Data Points:	Away Score Competitor Qualifier Ground Neutral Home Score Period Number Period Type Sport Event Id Sport Event Replaced By	Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining

```
curl -X GET "https://api.sportradar.com/basketball-probabilities/trial/v2/en/sport_events/sr:spor
t_event:26109208/sport_event_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball-probabilities/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /sport_event_probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Upcoming Probabilities

Provides a list of IDs for upcoming sport events in the next 24 hours.

TTL / Cache:	**1 second**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Upcoming Id

```
curl -X GET "https://api.sportradar.com/basketball-probabilities/trial/v2/en/sport_events/upcomin
g_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Upcoming Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/upcoming_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Timeline Probabilities

Provides a timeline of pre-match and live probability changes for a given game.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group - Group Name Group Id Group Name	Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Season Competition Id Season End Date Season Id Season Name	Season Start Date Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Division Gender	Id Name State
Player Info Data Points:	Id	Name	

Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score	Market Home Score Market Last Updated Market Live	Market Name Market Removed Outcome Name
Match Info Data Points:	Away Score Competitor Qualifier Ground Neutral Home Score Period Number Period Type Sport Event Id Sport Event Replaced By	Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining
Competitor Match Statistics Data Points:	Assists Ball Possession Biggest Lead Defensive Rebounds Fouls Free Throw Attempts Successful Free Throw Attempts Made Leader Assists Leader Assists Player Id	Leader Points Leader Points Player Id Leader Rebounds Leader Rebounds Player Id Offensive Rebound Rebounds Shots Blocked Steals Team Leads Team Rebounds	Team Turnovers Three Point Attempts Successful Three Point Attempts Total Time Spent in Lead Timeouts Turnovers Two Point Attempts Successful Two Point Attempts Total
Player Match Statistics Data Points:	Assists Blocks Defensive Rebounds Field Goals Attempted Field Goals Made Free Throws Attempted	Free Throws Made Minutes Offensive Rebounds Personal Fouls Points Steals	Technical Fouls Three Pointers Attempted Three Pointers Made Total Rebounds Turnovers

```
curl -X GET "https://api.sportradar.com/basketball-probabilities/trial/v2/en/sport_events/sr:spor
t_event:26109208/timeline_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Timeline Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/basketball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/timeline_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description

`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Hungarian (hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Swedish (sv), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Frequently Asked Questions

Q: What leagues or tournaments do you cover for basketball?

You can find all the leagues we cover, as well as a breakdown of data offered, via our Coverage Matrix.

Q: What format are date fields presented in?

A: When we present date only values we present these in the ISO 8601 standard format.
ex: 2013-04-03
We use these for attributes that have date and no time (such as birthdate). For more information:
https://en.wikipedia.org/wiki/ISO_8601

Q: What format are the date/time fields presented in?

A: All of our Date/Time attributes are in UTC, presented in the ISO 8601 standard format.
ex: 2013-04-03T18:15:00+00:00
For more information: https://en.wikipedia.org/wiki/ISO_8601

Q: How can I find the values for various enum data points within the API?

A: Many enum values are listed in the FAQ below. For the most up-to-date values, please see the Schema section of the OpenAPI specification here: https://api.sportradar.com/basketball/trial/v2/openapi/swagger/index.html

Q: What are the valid sport_event_status - status values?

A: Here are the valid sport_event_status - status values and their definitions:

- not_started – The game is scheduled to be played
- started – The game has begun
- live – The game is currently in progress

- postponed – The game has been postponed to a future date
- suspended - The game began, but has been suspended to a later time
- match_about_to_start – The game is about to begin
- delayed – The game has been temporarily delayed and will be continued
- interrupted - The game began, but coverage has stopped for a short time. Note that game scores may not be updated during this period, the last recorded game score will be displayed instead
- cancelled – The game has been cancelled and will not be played
- ended – The game is over
- closed – The game results have been confirmed

Q: What are the valid sport_event_status – match_status values?

A: Here are the valid sport_event_status - status values:

- 1st_quarter
- 2nd_quarter
- 3rd_quarter
- 4th_quarter
- 1st_half
- 2nd_half
- overtime
- penalties
- pause
- awaiting_extra_time
- awaiting_penalties
- interrupted
- not_started
- aet
- ended
- postponed
- cancelled
- abandoned
- start_delayed
- started

Q: How do I access past seasons results and stats?

A: Use the Competition Seasons endpoint to locate the season_id for the season you want to access. Use that season_id to interrogate the various seasons endpoints.

Q: What are the valid outcomes for probabilities?

A: Here are the valid outcome probabilities:

- home_team_winner
- away_team_winner

Q: How do I find out the coverage for a particular match?

A: Find the node called "coverage" in the Summary, Lineups, or Timeline feeds. The attribute "coverage - live" reports if Sportradar has live coverage of the match or not. Properties within the coverage node denotes specific coverage properties, and displays a boolean value if the coverage type is available. A list of possible properties and their definitions are listed below:

- boxscore - Player statistics are available post-match.
- lineups - Lineups are available, with starting players, pre-match.
- match_clock - A running match clock is available.
- stats_leaders - Live statistics leaders are available. leader_points, leader_assists, leader_rebounds.
- time_played_minutes_and_seconds - Player statistic for minutes is available in minutes and seconds.
- time_played_minutes_only - Player statistic for minutes is available however only in minutes.

Q: Why do different groups have coverage information for a competition?

A: The notion of “groups” is versatile and is used to distinguish between playoffs, and our competition structures and is therefore necessary to describe coverage at a group level for consistency. Generally, however, within a competition in Basketball, there will be no difference between competition coverage between conferences.

Q: What are the possible event type values?

A: Here are the possible values for an event type:

- match_started
- period_start
- rebound
- score_change
- foul
- ball_block
- break_start
- timeout
- period_score
- match_ended
- attempt_missed
- free_throws_awarded
- steal
- timeout_over
- turnover
- video_review
- video_review_over
- won_jump_ball

Q: What are markets and what are the different markets?

A: Markets is something you can bet on that we provide probabilities for. Over time we intend to provide more and more markets in the API. Currently the only market we provide is 2-way (will the home team win? Or the away team?).

Q: What is the scale of the X Y coordinates?

A: The court we use is 100 by 100. Here is a layout of the court:

0		***X***		**100**
Y				
100				

Q: How are group IDs delivered in the stage array with the various types?

A: With the type of "league" they will have a sr:league prefix. With the type of "cup" they will have a sr:cup prefix.

Q: Do you provide fixtures for future matches, before the competitors are known?

A: For playoff phase matches in selected leagues*, we will provide fixtures before the competitors are known. These matches will be created with placeholder competitors and when the fixture is fully drawn, the correct competitors will overwrite the placeholder names, with the match ID remaining the same.

Regarding the competitor placeholder names, we look to follow official naming where possible. This does mean that some differences will apply across competitions.

* Selected leagues are listed below:

- Euroleague
- Eurocup
- Pro A
- Liga ACB
- BBL
- TBSL
- United League
- A1
- Liga ABA
- NBL
- Israel Super League
- CBA

Q: How are “live” feeds handled in the API?

A: Sport Events appear in the feed 10 minutes before the scheduled start time and are removed 10 minutes after the Sport Event is ended.

Q: What is the probabilities package?

A: The probabilities package is an add-on set of feeds that are an extension of the Season Probabilities feed in the main package (which already provides pre-match probabilities for the sport event winner market). The main features of the

Probabilities extension are: Live Probabilities that update throughout game, Season Outright Probabilities for the Tournament Winner market, and Live Probabilities Coverage indicator for the next 24 hours.

Q: Is there Live Probability coverage for every game you cover in the Global Basketball API?

A: Live Probability coverage depends on a number of factors and we can't guarantee that every game will be covered for any given league, however, Sportradar aims to cover close to 100% of games from the following Tier 1 league: National Basketball Association (NBA) – USA

Q: Can I utilize the Probabilities extension if I license the Sportradar NBA API?

A: If you have a license for our NBA API you can also integrate Live Probabilities by utilizing the ID Mapping feeds available in Global Basketball v2 to connect games between the two products.

Q: What are the possible standings types I can observe in the Standings feed?

A: Listed below are the values for standing – type.

- total
- home
- away
- first_half_total
- first_half_home
- first_half_away
- second_half_total
- second_half_home
- second_half_away

Q: Why can't I find a particular match in the Daily Summaries, Season Summaries or Sport Events Updated feeds?

A: These endpoints support pagination and return 200 entries by default. To return more matches, an additional query string parameter must be used after your API key. For example, appending *&start=200* will return the next 200 entries per page, *&start=400* will return the next 200, and so on.

Q: How will a sport event behave when it is not covered with live scores?

A: When a sport_event is not covered live, the status and match_status will remain as not_started until results are entered post-match.

Q: How do I locate the TTL (Time to Live)/cache on an API endpoint?

A: The cache (in seconds) can be accessed in the returned header information on each RESTful API call, under `cache-control`.
ex. `cache-control: max-age=1, public, s-maxage=1` or
`cache-control: public, must-revalidate, max-age=120`

Q: What are the possible values for `cup_round` – `state` in the Season Links feed?

A: Listed below are the values and definitions for `cup_round` - `state`. These can be leveraged to determine the status of a cup round.

- `empty` - A matchup has been created but neither the match details nor the competitors are known.
- `unseeded_fixture` - Match details are known but competitors are unknown.
- `partial_seeded` - One competitor is known.
- `partial_seeded_fixture` - Match details and one competitor are known.
- `seeded` - Both competitors are known.
- `seeded_fixture` - Match details and both competitors are known.
- `unstarted` - Match(es) have been added.
- `on_going` - The first match has started.
- `decided` - The last match has ended.
- `winner` - The winner is known.
- `cancelled` – The matchup has been cancelled.

Return to top

SPORTRADAR
150 South 5th St. Suite 400
Minneapolis, MN 55402
USA

us-sales@sportradar.com
us-jobs@sportradar.com
Main: 612-361-4100
Sales: 612-361-4102
Terms and Conditions

SPORTRADAR IS SOCIAL

Exhibit 55

sportradar

⁝ Docs Navigation ☰ Menu

Bash Ruby Python

Global American Football v2

Statistics Summary

Documentation

OpenAPI Specification

Change Log

Coverage Matrix

Frequently Asked Questions

Quick Access	
Postman Collection	**XSD Schema**
▶ Run in Postman	Download

Global American Football API Overview

The Global American Football API provides real-time game updates (when available) and a database of supplementary statistics. Data is collected via Sportradar on-venue scouts and in-house operators.

Leagues covered include NFL, NCAA, CFL, European League football, and more. Select the **Global American Football** package in our Coverage Matrix for competitions and data offered.

The API is consistent in structure, format, and behavior with the other General Sport APIs. Primary feeds will return seasons, competitions, team and player data, and real-time scores.

Additional feeds provide a host of complementary stats, including:

- Standings
- Player profiles
- Team profiles
- Historical results
- Match win probabilities

An extended Probabilities package is also offered. This add-on includes in-game probability updates and season outrights.

API	API Version

Global American Football	v2

Note: Authentication is required for all API calls.

Global American Football API Map

To best utilize the Global American Football API, you will need several parameters to create your API calls. The map below illustrates how you can obtain the parameters you need.

Example:

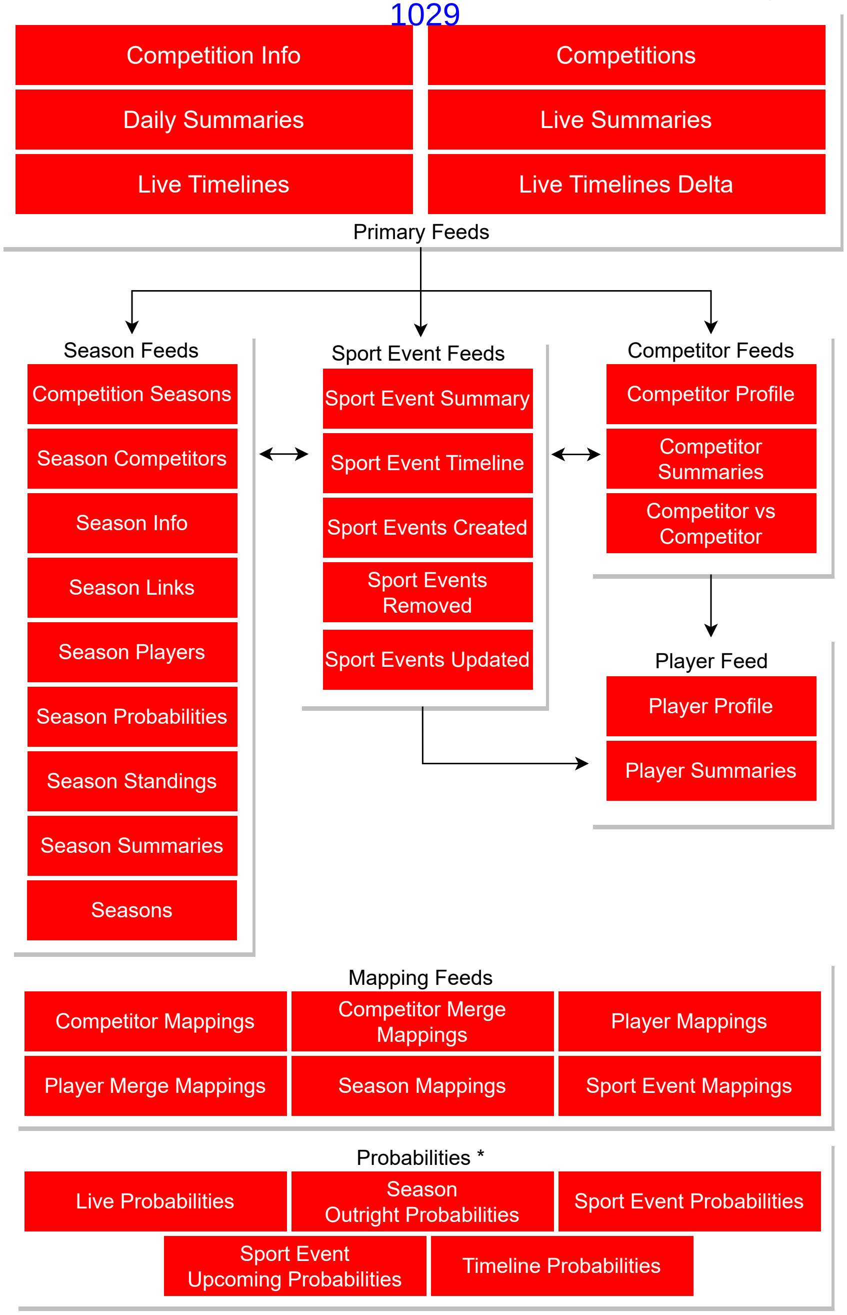
To find the probability of a home team win for a given game:

1. Call the Daily Schedule for the day the sport event takes place and find the Season Id for the chosen sport event
2. Make note of the Sport Event Id for the given sport event
3. Call the Season Probabilities using the Season Id
4. Find Sport Event Id within the results locate the home team and the outcome probability

The probability of a home team win is displayed.

The primary feeds require only a date or human-readable parameters to call the endpoints. Those feeds provide Sport Event Ids, Competitor Ids, or Season Ids which can be used to generate the match, team, and tournament feeds.

- - Available only with the probabilities plan within the Global American Football v2 package. See FAQ for details.



Competition Info

Provides the name, id, and parent id for a given competition.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Gender Id	Name Parent Id	State

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitions/sr:competition:3
1/info.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/competitions/`{competition_id}`/info.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Download

Return to top

Competition Seasons

Provides historical season information for a given competition. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitions/sr:competition:3
1/seasons.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /competitions/ `{competition_id}` /seasons. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitions

Provides a list of all available competitions.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Gender Id	Name State	Parent Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitions.xml?api_key={you
r_api_key}"
```

The above command returns xml structured like this.

Access the Competitions feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /competitions. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Mappings

Provides team id mapping between the Global American Football API and other football APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 competitor mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitors/mappings.xml?api_
key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/competitors/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/competitors/mappings.xml?api_key={you
r_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Competitor Merge Mappings

Provides the valid Sportradar Id in cases when two competitors have been merged into one. Mapping entries will remain in the feed for 7 days.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	Name	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/americanfootball/trial/v2/en/competitors/merge_mappings.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/competitors/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Profile

Provides top-level information for a given team, including the full team roster, home venue, and team colors.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name Competitor State	Jersey Base Color Jersey Horizontal Stripes Jersey Number Color Jersey Shirt Type Jersey Sleeve Color	Jersey Sleeve Detail Jersey Split Jersey Squares Jersey Stripes Jersey Type
Player Info Data Points:	Country Code Date of Birth Gender Height	Id Jersey Number Nationality	Name Position Weight

Venue Info Data Points:	Capacity City Country Code	Country Name Id	Map Coordinates Name

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitors/sr:competitor:442
4/profile.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Summaries

Provides previous and upcoming game information for a given competitor, including results for past games and scheduling info for upcoming games.

TTL / Cache:		**300 seconds**	
Update Frequency:		As Necessary	
Content Type:		XML or JSON	
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id	Season Year Stage End Date Stage Order Stage Phase Stage Start Date

	Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Season End Date Season Id Season Name Season Start Date	Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Country	Country Code Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number Period Type Sport Event Id	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitors/sr:competitor:4424/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor vs Competitor

Provides previous and upcoming games between two teams including scoring information.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number Period Type Sport Event Id	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/competitors/sr:competitor:442
4/versus/sr:competitor:4324/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor vs Competitor feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /versus/ `{competitor2_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`competitor2_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Daily Summaries

Provides game information for a given day including scoring information.

TTL / Cache:		**300 seconds**	
Update Frequency:		As Necessary	
Content Type:		XML or JSON	
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score

	Period Type	Normaltime Score	Sport Event Status Match Status
	Sport Event Id	Sport Event Status Away Score	Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/schedules/2022-01-09/summarie
s.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Daily Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /schedules/ `{year}` - `{month}` - `{day}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`year`	Year in 4 digit format (YYYY).
`month`	Month in 2 digit format (MM).
`day`	Day in 2 digit format (DD).
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/schedules/2022-01-09/summaries.xml?ap
i_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 200.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description

`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Live Summaries

Provides game information for all currently live games, including team scoring. This feed updates in real time as games are played.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number Period Type Sport Event Id	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/schedules/live/summaries.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Live Timelines

Provides a play-by-play event timeline for currently live games.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Away Score Period Home Score Period Score Number Period Score Type Sport Event Status	Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score	Sport Event Status Match Status Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time
Timeline Info Data Points:	Away Score Break Name Competitor Home Score	Id Match Clock Match Time Period	Period Name Time Type

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/schedules/live/timelines.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Live Timelines Delta

Provides a 10 second live delta of game information, including scoring and a play-by-play event timeline.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Away Score Period Home Score Period Score Number Period Score Type Sport Event Status	Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score	Sport Event Status Match Status Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time
Timeline Info Data Points:	Away Score Break Name Competitor Home Score	Id Match Clock Match Time Period	Period Name Time Type

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/schedules/live/timelines_delta.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines Delta feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /schedules/live/timelines_delta. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Mappings

Provides player id mapping between the Global American Football API and other football APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 player mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/players/mappings.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Player Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /players/mappings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/players/mappings.xml?api_key={your_ap
i_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	**Description**
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Merge Mappings

Provides valid ids for players who have had their profiles merged. While Sportradar always strives to provide one unique player id, it is a possibility for two ids to be created. This feed provides the correct id once profiles have been merged.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mapping Info Data Points:	Merged Id	Name	Retained Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/players/merge_mappings.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/players/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/players/merge_mappings.xml?api_key={y
our_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Profile

Provides player biographical information, including current team.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Player Info Data Points:	Active Country Code Date Of Birth End Date Gender	Height Id Jersey Number Nationality Name	Name Position Role Type Start Date Weight

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/players/sr:player:834683/profile.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`player_id`	ID of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Summaries

Provides game info and results for the past 10 games in which a given player participated.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number Period Type Sport Event Id	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/players/sr:player:834683/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages

	Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`player_id`	ID of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Competitors

Provides a list of teams participating for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Id	Name	Short Name

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/compe
titors.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Competitors feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /competitors. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Info

Provides detailed information for a given season, including participating teams and league structure.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Group - Group Name Group Id Group Max Rounds Group Name Info Competition Status	Info Venue Reduced Capacity Info Venue Reduced Capacity Max Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/info.
xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /info. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Links

Provides information about linked cup rounds for a given season.

Use this feed to compile full advancement brackets for relevant seasons/tournaments. Links between all matches and rounds are available when competitors (TBD vs. TBD) are not yet known.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Competition Id Cup Round Id Cup Round Name Cup Round Order Cup Round State Cup Round Type Cup Round Winner Id Group - Group Name	Group Id Stage End Date Stage Phase	Stage Start Date Stage Type Stage Year
Match Info Data Points:	Id Country	Start Time Id	Start Time Confirmed State

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/stage
s_groups_cup_rounds.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Links feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /stages_groups_cup_rounds. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Mappings

Provides season id mapping between the Global American Football API and other football APIs.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/mappings.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Season Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/mappings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/seasons/mappings.xml?api_key={your_ap
i_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Season Players

Provides names and ids for all participating players for a given season.

TTL / Cache:	300 seconds		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Player Info Data Points:	First Name Display First Name	Display Last Name Id	Last Name Name

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:74885/players.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Players feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /players. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.

`your_api_key`	Your API key.

Return to top

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:74885/players.xml?a
pi_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Season Probabilities

Provides 2-way win probabilities (home team win, away team win) for all games for a given season.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

	Round - Cup Round Number of Sport Events	Season Name Season Start Date	
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Home Team Win Probability Home Team Win Probability	Market Name	Outcome Name
Match Info Data Points:	Competitor Qualifier Sport Event Id	Sport Event Start Time	Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/proba
bilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/americanfootball/trial/v2/en/seasons/sr:season:52800/probabilitie
s.xml?api_key={your_api_key}&start=200'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Season Standings

Provides detailed standings info for a given season.

TTL / Cache:	**10 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Group - Group Name Group Live Group Id Group Name	Stage End Date Stage Order Stage Phase	Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Standings Info Data Points:	Change Current Outcome Draw Goals Against Goals Difference Goals For	Group - Group Name Group Id Group Live Group Name Losses Played	Points Rank Season Standings Round Season Standings Tie Break Rule Season Standings Type Winning Percentage Wins

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/stand
ings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Standings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /standings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Summaries

Provides schedule information and scoring for all matches from a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 500 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed	Country Name Id	Name Reduced Capacity

	City Country Code	Map Coordinates	Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number Period Type Sport Event Id	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/summa
ries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/seasons/sr:season:52800/summaries.xm
l?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Seasons

Provides a list of historical season information for all competitions. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/seasons.xml?api_key={your_api
_key}"
```

The above command returns xml structured like this.

Access the Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /seasons. `{format}` ?
api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Mappings

Provides game id mapping between the Global American Football API and other football APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 sport event mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/mappings. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Event Summary

Provides real-time scoring for a given game. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score	Sport Event Start Time Sport Event Start Time	Sport Event Status Decided Overtime

	Period Home Score Period Number Period Type Sport Event Id	Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/sport_events/sr:match:3158803
1/summary.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Summary feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{match_id}` /summary. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`match_id`	ID of a given match.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Timeline

Provides real-time scoring and a play-by-play event timeline for a given game. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id	Round - Cup Round Sport Event Number Round Name Round Number	Season Year Stage End Date Stage Order Stage Phase

	Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name State
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Period Away Score Period Home Score Period Number Period Type Sport Event Id	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score	Sport Event Status Decided Overtime Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id
Timeline Info Data Points:	Away Score Break Name Competitor Home Score	Id Match Clock Match Time Period	Period Name Time Type

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/sport_events/sr:match:31588031/timeline.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /timeline. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given sport event.
`format`	xml or json.

`your_api_key`	Your API key.

Return to top

Sport Events Created

Provides ids for sport events that have been created in the last 24 hours.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Active Season Flag	Id	Created At

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/sport_events/created.xml?api_
key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Created feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/sport_events/created.`{format}`?api_key=`{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/sport_events/created.xml?api_key={you
r_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Removed

Provides ids for sport events that have been removed from the API due to an entry error. Ids will remain in the response for 2 weeks.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Removed Id

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/sport_events/removed.xml?api_
key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Removed feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/sport_events/removed.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en)

	Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/sport_events/removed.xml?api_key={you
r_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Updated

Provides ids for sport events that have been updated in the last 24 hours.

TTL / Cache:	**60 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Match Info Data Points:	Id	Updated At

```
curl -X GET "https://api.sportradar.us/americanfootball/trial/v2/en/sport_events/updated.xml?api_
key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Updated feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball/`{access_level}`/`{version}`/`{language_code}`/sport_events/updated.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/americanfootball/trial/v2/en/sport_events/updated.xml?api_key={you
r_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Probabilities Feeds

This collection of probability feeds is available for licensing as an add-on feature. They are an extension of the Season Probabilities feed in the main package which provides pre-match probabilities for the sport event winner market.

The main features of the Probabilities extension (where coverage applies) are:

- Live Probabilities that update throughout game
- Season Outright Probabilities for the Tournament Winner market
- Live Probabilities Coverage indicator for the next 24 hours

For more information or to request a Trial of this feature, please contact a Sportradar Sales representative.

The Open API specification and Syntax for using our Probabilities feeds, including examples of the payloads can be found below.

OpenAPI Specification

Live Probabilities

Provides top-level information for live games. If probabilities are available for a game, pre-match and live probabilities will be displayed.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country Country Code	Division Gender Id Name	State Virtual
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score Market Home Score	Market Last Updated Market Live Market Name Market Removed	Match Time Outcome Name Remaining Time Remaining Time in Period
Match Info Data Points:	Away Score Competitor Qualifier	Sport Event Resumed Time Sport Event Start Time	Sport Event Status Home Normaltime Score

	Home Score Period Number Period Type Sport Event Id Sport Event Replaced By	Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Overtime	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id Time Played Time Remaining

```
curl -X GET "https://api.sportradar.us/americanfootball-probabilities/trial/v2/en/schedules/live/
probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball-probabilities/`{access_level}`/`{version}`/`{language_code}`/schedules/live/probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Outright Probabilities

Provides a list of outright probabilities for each competitor from a given season.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Age Group Country Country Code	Division Gender Id Name	State Virtual

Probability Info Data Points:	Win Probability	Market Name	Outcome Name

```
curl -X GET "https://api.sportradar.us/americanfootball-probabilities/trial/v2/en/seasons/sr:seas
on:75095/outright_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Outright Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball-probabilities/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /outright_probabilities. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Probabilities

Provides pre-match and live probabilities for a given match.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

	Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Season Id Season Name Season Start Date	
Competitor Info Data Points:	Abbreviation Age Group Country Country Code	Division Gender Id Name	State Virtual
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score Market Home Score	Market Last Updated Market Live Market Name Market Removed	Match Time Outcome Name Remaining Time Remaining Time in Period
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id Sport Event Replaced By	Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id Time Played Time Remaining

```
curl -X GET "https://api.sportradar.us/americanfootball-probabilities/trial/v2/en/sport_events/s
r:sport_event:25496754/sport_event_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball-probabilities/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.

`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Upcoming Probabilities

Provides a list of IDs for upcoming sport events in the next 24 hours.

TTL / Cache:	**1 second**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Upcoming Id

```
curl -X GET "https://api.sportradar.us/americanfootball-probabilities/trial/v2/en/sport_events/up
coming_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Upcoming Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/upcoming_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Timeline Probabilities

Provides a timeline of pre-match and live probability changes for a given game.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Group - Group Name Group Id Group Name Round - Cup Round Id Round - Cup Round Number of Sport Events	Round - Cup Round Sport Event Number Round Name Round Number Round - Other Sport Event Id Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country Country Code	Division Gender Id Name	State Virtual
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score Market Home Score	Market Last Updated Market Live Market Name Market Removed	Match Time Outcome Name Remaining Time Remaining Time in Period
Match Info Data Points:	Away Score Competitor Qualifier Home Score Period Number Period Type Sport Event Id Sport Event Replaced By	Sport Event Resumed Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Away Normaltime Score Sport Event Status Away Score Sport Event Status Decided Overtime	Sport Event Status Home Normaltime Score Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Winner Id Time Played Time Remaining

```
curl -X GET "https://api.sportradar.us/americanfootball-probabilities/trial/v2/en/sport_events/s
r:sport_event:25496754/timeline_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Timeline Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/americanfootball-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/timeline_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Frequently Asked Questions

Q: What leagues do you cover, and at what level?

Coverage information can be found via our Coverage Matrix.

Please note that `sr:competition:27653` (NCAA, Regular Season) includes DI, DII, and DIII games.

Q: What format are date fields presented in?

A: When we present date only values we present these in the ISO 8601 standard format.
ex: 2013-04-03
We use these for attributes that have date and no time (such as birthdate). For more information: https://en.wikipedia.org/wiki/ISO_8601

Q: What format are the date/time fields presented in?

A: All of our Date/Time attributes are in UTC, presented in the ISO 8601 standard format.
ex: 2013-04-03T18:15:00+00:00
For more information: https://en.wikipedia.org/wiki/ISO_8601

Q: How can I find the values for various enum data points within the API?

A: Many enum values are listed in the FAQ below. For the most up-to-date values, please see the Schema section of the OpenAPI specification here: https://api.sportradar.com/americanfootball/trial/v2/openapi/swagger/index.html

Q: What are the valid sport_event_status - status values?

A: Here are the valid sport_event_status values and their definitions:

- not_started – The match is scheduled to be played
- started – The match has begun
- live – The match is currently in progress
- postponed – The match has been postponed to a future date
- suspended - The match began, but has been suspended to a later time
- delayed – The match has been temporarily delayed and will be continued
- cancelled – The match has been cancelled and will not be played
- match_about_to_start – The match is about to begin
- interrupted - The match began, but coverage has stopped for a short time. Note that match scores may not be updated during this period, the last recorded match score will be displayed instead
- ended – The match is over
- closed – The match results have been confirmed

Q: What are the valid sport_event_status – match_status values?

A: Here are the valid match_status values:

- not_started
- ended
- postponed
- cancelled
- interrupted
- overtime
- abandoned
- 1st_quarter
- 2nd_quarter
- 3rd_quarter
- 4th_quarter
- pause
- awaiting_extra_time
- aet
- start_delayed

Q: What are the possible event types logged?

A: Here are the valid event type:

- match_started
- period_start
- score_change
- period_score
- break_start
- match_ended

Q: How do I access past seasons results and stats?

A: Use the Seasons endpoint to locate the season_id for the season you want to access. Use that season_id to interrogate the various seasons endpoints.

Q: What are the valid outcomes for probabilities?

A: Here are the valid outcome probabilities:

- home_team_winner
- away_team_winner

Q: How do I find out the coverage for a particular match?

A: Find the node called "coverage" in any of the Summary, Lineups, or Timeline feeds. The attribute "coverage - live" reports if Sportradar has live coverage of the match or not.

Q: Why do different groups have coverage information for a competition?

A: The notion of “groups” is versatile and is used to distinguish between playoffs, and our competition structures and is therefore necessary to describe coverage at a group level for consistency. Generally, however, within a competition in Football, there will be no difference between competition coverage.

Q: What are markets and what are the different markets?

A: Markets is something you can bet on that we provide probabilities for. Over time we intend to provide more and more markets in the API. Currently the only market we provide is 2-way (will the home team win? Or the away team?).

Q: How are group IDs delivered in the stage array with the various types?

A: With the type of "league" they will have a sr:league prefix. With the type of "cup" they will have a sr:cup prefix.

Q: How are “live” feeds handled in the API?

A: Sport Events appear in the feed 10 minutes before the scheduled start time and are removed 10 minutes after the Sport Event is ended.

Q: What is the probabilities package?

A: The probabilities package is an add-on set of feeds that are an extension of the Season Probabilities feed in the main package (which already provides pre-match probabilities for the sport event winner market). The main features of the Probabilities extension are: Live Probabilities that update throughout game, Season Outright Probabilities for the Tournament Winner market, and Live Probabilities Coverage indicator for the next 24 hours.

Q: Is there Live Probability coverage for every game you cover in the American Football API?

A: Live Probability coverage depends on a number of factors and we can’t guarantee that every game will be covered for any given league, however, Sportradar aims to cover close to 100% of games from the following Tier 1 league: National Football League (NFL) – USA

Q: Can I utilize the Probabilities extension if I license the Sportradar NFL API?

A: If you have a license for our NFL API you can also integrate Live Probabilities by utilizing the ID Mapping feeds available in American Football v2 to connect games between the two products.

Q: What are the possible standings types I can observe in the Standings feed?

A: Listed below are the values for standing – type.

- total
- home
- away
- first_half_total
- first_half_home
- first_half_away
- second_half_total
- second_half_home
- second_half_away

Q: Why can't I find a particular match in the Daily Summaries, Season Summaries or Sport Events Updated feeds?

A: These endpoints support pagination and return 200 entries by default. To return more matches, an additional query string parameter must be used after your API key. For example, appending *&start=200* will return the next 200 entries per page, *&start=400* will return the next 200, and so on.

Q: How will a sport event behave when it is not covered with live scores?

A: When a sport_event is not covered live, the status and match_status will remain as not_started until results are entered post-match.

Q: How do I locate the TTL (Time to Live)/cache on an API endpoint?

A: The cache (in seconds) can be accessed in the returned header information on each RESTful API call, under `cache-control`. ex. `cache-control: max-age=1, public, s-maxage=1` or `cache-control: public, must-revalidate, max-age=120`

Q: What are the possible values for `cup_round` – `state` in the Season Links feed?

A: Listed below are the values and definitions for `cup_round` - `state`. These can be leveraged to determine the status of a cup round.

- `empty` - A matchup has been created but neither the match details nor the competitors are known.
- `unseeded_fixture` - Match details are known but competitors are unknown.
- `partial_seeded` - One competitor is known.
- `partial_seeded_fixture` - Match details and one competitor are known.
- `seeded` - Both competitors are known.
- `seeded_fixture` - Match details and both competitors are known.
- `unstarted` - Match(es) have been added.
- `on_going` - The first match has started.
- `decided` - The last match has ended.
- `winner` - The winner is known.
- `cancelled` – The matchup has been cancelled.

Return to top

SPORTRADAR
150 South 5th St. Suite 400
Minneapolis, MN 55402
USA

us-sales@sportradar.com
us-jobs@sportradar.com
Main: 612-361-4100
Sales: 612-361-4102
Terms and Conditions

SPORTRADAR IS SOCIAL

Facebook Technology Partner

Exhibit 56

sportradar

⋮ Docs Navigation ☰ Menu

Bash Ruby Python

Global Ice Hockey v2

Statistics Summary

Documentation

OpenAPI Specification

Change Log

Coverage Matrix

Frequently Asked Questions

Quick Access	
Postman Collection	**XSD Schema**
▶ Run in Postman	Download

Global Ice Hockey API Overview

The Global Ice Hockey API provides real-time scoring and statistics (when available) and an array of supplementary data. Data is collected via Sportradar's on-venue scouts and in-house operators.

80+ leagues are covered, including: SHL, Liiga, Czech Extraliga, Ekstraliga, HKL, and Olympics Men. Select the **Global Ice Hockey** package in our Coverage Matrix for competitions and data offered.

The API is consistent in structure, format and behavior with the other General Sport APIs. Primary feeds will return seasons, competitions, team and player data, and real-time scores.

Additional feeds provide a host of complementary stats, including:

- Standings
- Player profiles
- Team profiles
- Historical results
- Game lineups
- Seasonal statistics
- Match win probabilities

Real-time customers are also offered two delivery Push Feeds to enhance speed.

An extended Probabilities package is also offered. This add-on includes in-game probability updates and season outrights.

API	API Version
Global Ice Hockey	v2

Note: Authentication is required for all API calls.

Global Ice Hockey API Map

To best utilize the Global Ice Hockey API, you will need several parameters to create your API calls. The map below illustrates how you can obtain the parameters you need.

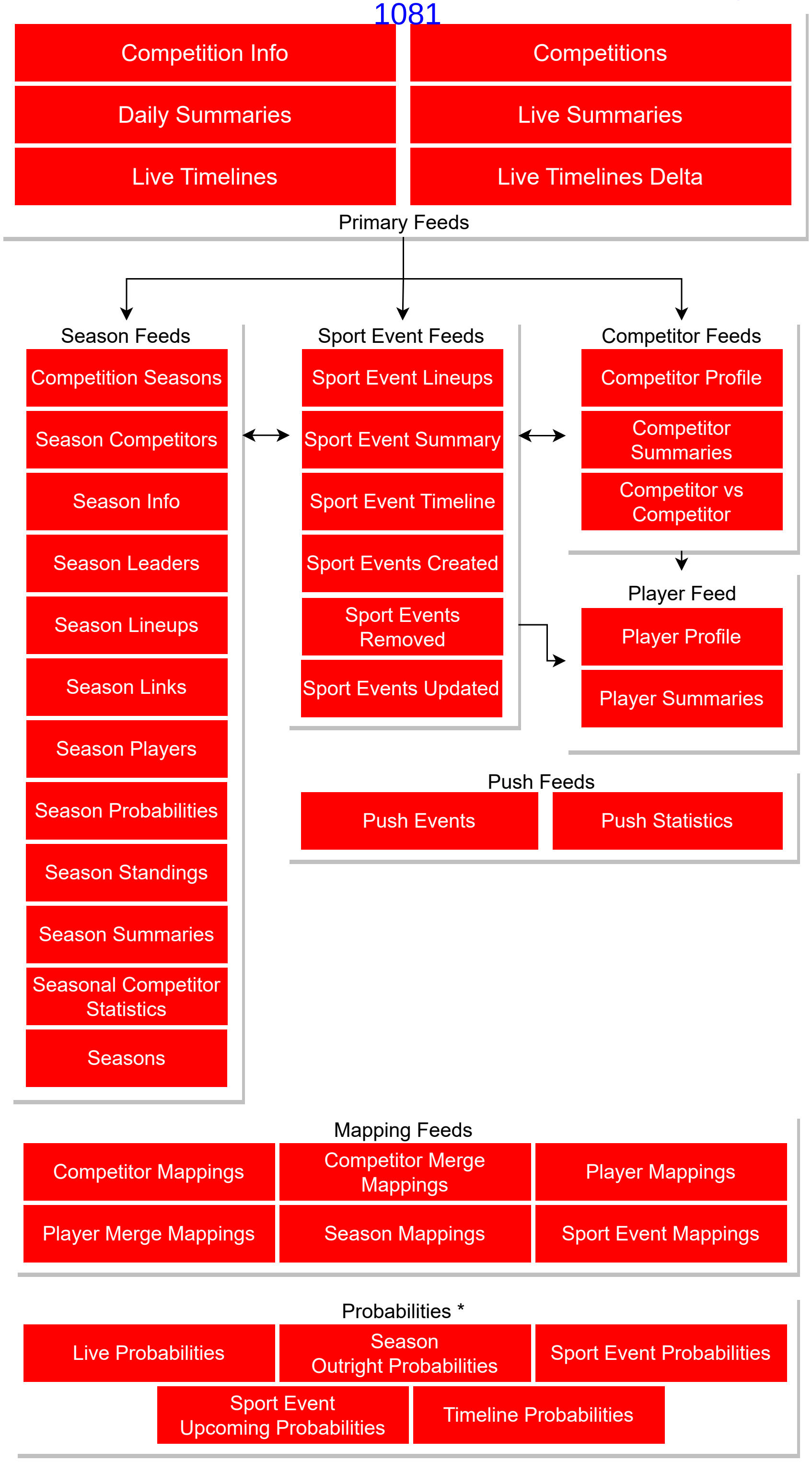
Example:

To find the probability of a home team win for a given game:

1. Call the Daily Schedule for the day the sport event takes place and find the Season Id for the chosen sport event
2. Make note of the Sport Event Id for the given sport event
3. Call the Season Probabilities using the Season Id
4. Find Sport Event Id within the results locate the home team and the outcome probability

The probability of a home team win is displayed.

The primary feeds require only a date or human-readable parameters to call the endpoints. Those feeds provide Sport Event Ids, Competitor Ids, or Season Ids which can be used to generate the match, team, and tournament feeds.



Competition Info

Provides the name, id, and parent id for a given competition.

TTL / Cache:	**300 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON

Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Competition Gender	Competition Id	Competition Name

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitions/sr:competition:3/info.
xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /competitions/ `{competition_id}` /info. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competition Seasons

Provides historical season information for a given competition. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitions/sr:competition:3/seaso
ns.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /competitions/ `{competition_id}` /seasons. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitions

Provides a list of all available competitions.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Competition Gender	Competition Id	Competition Name

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitions.xml?api_key={your_api_
key}"
```

The above command returns xml structured like this.

Access the Competitions feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /competitions. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Mappings

Provides team id mapping between the Global Ice Hockey API and other hockey APIs.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitors/mappings.xml?api_key={y
our_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /competitors/mappings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.

`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/competitors/mappings.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Competitor Merge Mappings

Provides the valid Sportradar Id in cases when two competitors have been merged into one. Mapping entries will remain in the feed for 7 days.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	Name	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitors/merge_mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/competitors/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Profile

Provides top-level information for a given team, including the full team roster, home venue, and team colors.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name	Jersey Base Color Jersey Horizontal Stripes Jersey Number Color Jersey Shirt Type Jersey Sleeve Color	Jersey Split Jersey Squares Jersey Stripes Jersey Type
Player Info Data Points:	Country Code Date of Birth Gender Height	Id Jersey Number Nationality	Name Position Weight
Venue Info Data Points:	Capacity City Country Code	Country Name Id	Map Coordinates Name

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitors/sr:competitor:3696/prof
ile.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/competitors/`{competitor_id}`/profile.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor Summaries

Provides previous and upcoming game information for a given competitor, including results for past games and scheduling info for upcoming games.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Live Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	

Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Away Score Period Home Score Period Number Period Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Away Score Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitors/sr:competitor:3691/summ
aries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/competitors/`{competitor_id}`/summaries.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.

`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Competitor vs Competitor

Provides previous and upcoming games between two teams, including scoring information.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Live Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Away Score Period Home Score Period Number Period Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/competitors/sr:competitor:3691/vers
us/sr:competitor:3703/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor vs Competitor feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /versus/ `{competitor2_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`competitor2_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Daily Summaries

Provides game information for a given day including team scoring, player and team game statistics.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**300 seconds**

Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Live Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Away Score Period Home Score Period Number Period Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/schedules/2021-10-30/summaries.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Daily Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/schedules/`{year}`-`{month}`-`{day}`/summaries.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`year`	Year in 4 digit format (YYYY).
`month`	Month in 2 digit format (MM).
`day`	Day in 2 digit format (DD).
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/schedules/2019-01-15/summaries.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This will often be required to access all available data in this feed. By default the limit is set to 200.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Live Summaries

Provides game information for all currently live games, including team scoring. This feed updates in real time as games are played.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Live Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Away Score Period Home Score Period Number Period Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/schedules/live/summaries.xml?api_ke
y={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`competitor_id`	ID for a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Live Timelines

Provides a play-by-play event timeline for currently live games.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id	Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status	Sport Event Status Scout Abandoned Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time
Timeline Info Data Points:	Assist Player Id Assist Player Name Assist Type	Competitor Home Score Id	Period Period Name Suspension Minutes

	Away Score Break Name	Match Clock Match Time	Time Type

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/schedules/live/timelines.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Live Timelines Delta

Provides a 10 second live delta of game information, including scoring and a play-by-play event timeline.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Period Score Away Score Period Score Home Score Period Score Number Period Score Type Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id	Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status	Sport Event Status Scout Abandoned Sport Event Status Winner Id Sport Event Timeline Id Sport Event Timeline Start Time
Timeline Info Data Points:	Assist Player Id Assist Player Name Assist Type	Competitor Home Score Id	Period Period Name Suspension Minutes

	Away Score Break Name	Match Clock Match Time	Time Type

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/schedules/live/timelines_delta.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timelines Delta feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines_delta.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Mappings

Provides player id mapping between the Global Ice Hockey API and other hockey APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 player mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/players/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/players/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/players/mappings.xml?api_key={your_api_key}&
start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This may be required to access all available data in this feed. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Merge Mappings

Provides valid ids for players who have had their profiles merged. While Sportradar always strives to provide one unique player id, it is a possibility for two ids to be created. This feed provides the correct id once profiles have been merged.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mapping Info Data Points:	Merged Id	Name	Retained Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/players/merge_mappings.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Player Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/players/merge_mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/players/merge_mappings.xml?api_key={your_api
_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description

`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Player Profile

Provides player biographical information, including current team.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Country Code Date Of Birth Gender Height	Id Jersey Number Nationality	Name Position Weight

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/players/sr:player:29663/profile.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Profile feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`player_id`	ID of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Player Summaries

Provides game info, results, and statistics for the past 10 matches in which a given player participated.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Live Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Away Score Period Home Score Period Number Period Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number

Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/players/sr:player:29663/summaries.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`player_id`	Id of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Competitors

Provides a list of teams participating for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation	Competitor Id Competitor Name	Competitor Short Name

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/competitor
s.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Competitors feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /competitors. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Info

Provides detailed information for a given season, including participating teams and league structure.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Group Id Group Max Rounds Group Name Info Competition Status Info Venue Reduced Capacity	Info Venue Reduced Capacity Max Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/info.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Info feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /info. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Leaders

Provides a list of leaders for a given season. Statistics include goals, points, and assists.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation	Competitor Id	Competitor Name
Player Info Data Points:	Player Id	Player Name	Multiple Competitors Flag
Season Leader Info Data Points:	Data Point Type Data Point Value	Leader Rank	List Type

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:85082/leaders.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Leaders feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /leaders. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Lineups

Provides game rosters for all games in a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**30 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Coverage Live Coverage Property Type Coverage Property Value Group - Group Name Group Id	Group Name Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name	Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Split Jersey Squares Jersey Stripes Jersey Type	Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name Manager Nationality Manager Nickname

	Jersey Base Color Jersey Horizontal Stripes		
Player Info Data Points:	Country Code Date of Birth Height Id	Jersey Number Nationality Name Played	Position Starter Weight
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Score Away Score Period Score Home Score Period Score Number Period Score Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/lineups.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Lineups feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /lineups. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/seasons/sr:season:78917/lineups.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the lineup information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Season Links

Provides information about linked cup rounds for a given season.

Use this feed to compile full advancement brackets for relevant seasons/tournaments. Links between all matches and rounds are available when competitors (TBD vs. TBD) are not yet known.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Cup Round Id Cup Round Name Cup Round Order Cup Round State Cup Round Type Cup Round Winner Id Group - Group Name	Group Id Stage End Date Stage Phase	Stage Start Date Stage Type Stage Year
Match Info Data Points:	Sport Event Id	Sport Event Start Time	Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/stages_grou
ps_cup_rounds.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Links feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /stages_groups_cup_rounds. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Mappings

Provides season id mapping between the Global Ice Hockey API and other hockey APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 season mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/mappings.xml?api_key={your_
api_key}"
```

The above command returns xml structured like this.

Access the Season Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/mappings. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/seasons/mappings.xml?api_key={your_api_key}&
start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Season Players

Provides names and ids for all participating players for a given season.

Note: Pagination may be required to pull all data within this feed. By default, the feed will return 1,000 players.

TTL / Cache:	**300 seconds**

Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Player Info Data Points:	Display First Name Display Last Name First Name	Player Id Last Name	Name Nickname

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/players.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Players feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/players.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/seasons/sr:season:78917/players.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Season Probabilities

Provides 3-way match win probabilities (home team win, away team win, draw) for all games for a given season.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Type Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Home Team Win Probability Home Team Win Probability	Market Name	Outcome Name
Match Info Data Points:	Competitor Qualifier Neutral Ground Sport Event Id	Sport Event Replaced By Sport Event Start Time	Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/probabiliti
es.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ {access_level} / {version} / {language_code} /seasons/ {season_id} /probabilities. {format} ?api_key= {your_api_key}

Parameter	Description
access_level	Defines the access level of your API key as Production (production) or Trial (trial).
version	Version number of the API you are accessing (Current Version: v2).
language_code	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
season_id	Id of a given season.
format	xml or json.
your_api_key	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/probabilities.xml?a
pi_key={your_api_key}&start=200'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
start	Number to start the list of results from. Example: start=0
limit	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Season Standings

Provides detailed standings info for a given season.

TTL / Cache:	10 seconds

Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Group - Group Name Group Live Group Id Group Name	Stage End Date Stage Order Stage Phase	Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Standings Info Data Points:	Change Current Outcome Draw Goals Against Goals Difference Goals For Loss	Loss Normal Time Loss Overtime Loss Shootout Played Points Rank Round	Tie Break Rule Type Win Win Normal Time Win Overtime Win Shootout

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/standings.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Standings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/standings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/standings.xml?api_k
```

```
ey={your_api_key}&round=1'
```

In addition to the URL parameters listed above, you can filter the Season Standings information with the following optional query string parameter.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

Parameter	Description
`round`	Round number expressed as: {round}. Example: round=1
`live`	Live standings expressed as a boolean value. Example: live=true

Return to top

Season Summaries

Provides schedule information for all matches from a given season including scoring and statistics at the match level.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Type Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	

Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Away Score Period Home Score Period Number Period Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:85082/summaries.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	ID of a given season.
`format`	xml or json.

`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/seasons/sr:season:78917/summaries.xml?api_ke
y={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Seasonal Competitor Statistics

Provides team seasonal statistics for a given team and season.

TTL / Cache:	**30 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Sport Id	Sport Name	
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Competitor Season Statistics Data Points:	Form Goals	Matches Played Matches Won	Saves Shots on Target Shutouts

	Goals Allowed Matches Lost	Number of Penalties Penalty Minutes	
Player Season Statistics Data Points:	Assists First Assists First Goals Goals	Last Goals Minutes Played Penalties Penalty Minutes	Plus Minus Points Saves Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons/sr:season:78917/competitor
s/sr:competitor:344158/statistics.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasonal Competitor Statistics feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/competitors/`{competitor_id}`/statistics.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`competitor_id`	ID of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Seasons

Provides a list of historical season information for all competitions. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/seasons.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/seasons.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Lineups

Provides detailed roster information for a given match.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Type Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id	Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Split Jersey Squares	Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name

	Competitor Name Jersey Base Color Jersey Horizontal Stripes	Jersey Stripes Jersey Type	Manager Nationality Manager Nickname
Player Info Data Points:	Country Code Date of Birth Height Id	Jersey Number Nationality Name Played	Type Starter Weight
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Score Away Score Period Score Home Score Period Score Number Period Score Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/sr:sport_event:1844974
8/lineups.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Lineups feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /lineups. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given match.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Mappings

Provides game id mapping between the Global Ice Hockey API and other hockey APIs.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 sport event mappings.

TTL / Cache:	**300 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Mapping Info Data Points:	External Id	Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/sport_events/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/sport_events/mappings.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the summaries with one or more of the following optional query string parameters. This may be required to access all available data in this feed. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Event Summary

Provides real-time match-level statistics for a given match. Including player stats, team stats, and scoring by period. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Type Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground	Sport Event Id Sport Event Replaced By	Sport Event Status Decided by Fed

	Period Score Away Score Period Score Home Score Period Score Number Period Score Type	Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/sr:sport_event:2827447
4/summary.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Summary feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /summary. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given match.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Timeline

Provides real-time match-level statistics and a play-by-play event timeline for a given match. This includes player stats, team stats, scoring by period, clock time for each recorded event. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Coverage Type Group - Group Name Group Id Group Name Round Name	Round Number Season Competition Id Season End Date Season Id Season Name Season Start Date Season Year	Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name
Player Info Data Points:	Id	Name	
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Match Info Data Points:	Competitor Qualifier Neutral Ground Period Score Away Score Period Score Home Score Period Score Number Period Score Type	Sport Event Id Sport Event Replaced By Sport Event Start Time Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score	Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id Time Played Time Remaining
Timeline Info Data Points:	Assist Player Id Assist Player Name Assist Type Away Score Break Name	Competitor Home Score Id Match Clock Match Time	Period Period Name Suspension Minutes Time Type
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number

Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/sr:sport_event:2827447
4/timeline.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /timeline. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`sport_event_id`	ID of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Events Created

Provides ids for sport events that have been created in the last 24 hours.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Match Info Data Points:	Active Season Flag	Id	Created At

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/created.xml?api_key={y
our_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Created feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/sport_events/created.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/sport_events/created.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Removed

Provides ids for sport events that have been removed from the API due to an entry error. Ids will remain in the response for 2 weeks.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Removed Id

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/removed.xml?api_key={y
our_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Removed feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/`{version}`/`{language_code}`/sport_events/removed.`{format}`?api_key=`{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/sport_events/removed.xml?api_key={your_api_k
ey}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	**Description**

`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Sport Events Updated

Provides ids for sport events that have been updated in the last 24 hours.

TTL / Cache:	**60 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Match Info Data Points:	Id	Updated At

```
curl -X GET "https://api.sportradar.com/icehockey/trial/v2/en/sport_events/updated.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Updated feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/ `{access_level}` / `{version}` / `{language_code}` /sport_events/updated. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/icehockey/trial/v2/en/sport_events/updated.xml?api_key={your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. By default the limit is set to 1000.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=75

Return to top

Push Feeds

To best utilize Push feeds, we have included code samples in Ruby, Java and Python which provide an example of a way you can consume the feeds. Using these samples will output the feeds content to STDOUT.
For Java, we have also provided a Stream Client to assist your integration.

Note: In the provided Java sample, replace "URL GOES HERE" with the desired Push feed URL.

```
package com.sportradar.http.stream.client;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class StreamClientTest {

    private StreamClient client;

    private static String SERVICE_URL = "<url goes="" here="">";

    @Before
    public void setup() {
        client = new StreamClient();
    }

    @After
    public void cleanup() {
        client.terminate();
    }

    @Test
    public void testStream() throws Exception {
        Handler handler = new ConsoleHandler();
        client.stream(SERVICE_URL, handler);
        System.out.println("Connecting....");
        Thread.sleep(1 * 60 * 1000);
        System.out.println("Disconnecting....");
    }

}
```

Some of our APIs include Push feeds that allow you to get updates as soon as they are available. Push API feeds automatically send JSON and XML payloads to you via a push service, and can dramatically reduce the number of calls you need to make to our RESTful API feeds. The structure of the Push feeds are similar to the structure of the corresponding RESTful API feed (i.e. Push Events and Push Statistics). The push service ensures reliable and efficient delivery of the most up to date information.

Our Push services are based on a HTTP publish/subscribe model. When making a call to the Push APIs, you "subscribe" to various data feeds provided by our service; whenever new content is available on one of those feeds, the server pushes that information out to your client. When no new information is available on the feed, a heartbeat message is sent every 5 seconds to keep the connection active. If you want to filter the results of the feeds, there are several optional query string parameters that can be applied to the API call. If left unfiltered, then all data for the feed is displayed (i.e. all events).

For your applications to accept data from our Push feeds, ensure that your application can:

- Can follow a HTTP redirect or use the location provided in the feeds header within one minute of your initial request.
- Can accept HTTP data transfer encoded as chunked.

Our Push service does not provide a "stateful session", there is no memory of what data has been sent previously. If you are disconnected from the Push session, you can use the RESTful API to catch up or recover from the disconnection.

Syntax for using our Push feeds and examples of the JSON and XML payloads can be found below.

Push Events

Provides real-time event updates for all live matches.

```
curl -L GET 'api.sportradar.com/icehockey/trial/stream/events/subscribe?api_key={your_api_key}'
```

The above command returns json like this.

Access the Push Events feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/stream/events/subscribe?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'https://api.sportradar.com/icehockey/trial/stream/events/subscribe?api_key={your_api
_key}&amp;format=json&amp;sport_event_id=sr:sport_event_id:18449748'
```

In addition to the URL parameters listed above, you can filter the Events information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

URL Parameters	Description
`channel`	Channel type expressed as: {channel_type}. Example: channel=icehockey
`competition_id`	Competition Id expressed as: {competition_id}. Example: competition_id=sr:competition:237
`event_id`	Event type expressed as: {event_type}. Example: event_id=score_change
`format`	Format type expressed as: {format}. Example: format=json
`season_id`	Season id expressed as: {season_id}. Example: season_id=sr:season:68178
`sport_event_id`	Sport event id expressed as: {sport_event_id}. Example: sport_event_id=sr:sport_event:18449748
`sport_id`	Sport id expressed as: {sport_id}. Example: sport_id=sr:sport:4

Push Statistics

Provides real-time team and player match-level statistics for all live matches.

```
curl -L "GET api.sportradar.com/icehockey/trial/stream/statistics/subscribe?api_key={your_api_ke
y}"
```

The above command returns json like this.

Access the Push Statistics feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey/`{access_level}`/stream/statistics/subscribe?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'https://api.sportradar.com/icehockey/trial/stream/statistics/subscribe?api_key={your_api_key}&amp;format=json&amp;sport_event_id=sr:sport_event_id:18449748'
```

In addition to the URL parameters listed above, you can filter the Statistics information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

URL Parameters	Description
`channel`	Channel type expressed as: {channel_type}. Example: channel=icehockey
`competition_id`	Competition Id expressed as: {competition_id}. Example: competition_id=sr:competition:237
`event_id`	Event type expressed as: {event_type}. Example: event_id=score_change
`format`	Format type expressed as: {format}. Example: format=json
`season_id`	Season id expressed as: {season_id}. Example: season_id=sr:season:68178
`sport_event_id`	Sport event id expressed as: {sport_event_id}. Example: sport_event_id=sr:sport_event:18449748
`sport_id`	Sport id expressed as: {sport_id}. Example: sport_id=sr:sport:4

Probabilities Feeds

This collection of probability feeds is available for licensing as an add-on feature. They are an extension of the Season Probabilities feed in the main package which provides pre-match probabilities for the sport event winner market.

The main features of the Probabilities extension (where coverage applies) are:

- Live Probabilities that update throughout game
- Season Outright Probabilities for the Tournament Winner market
- Live Probabilities Coverage indicator for the next 24 hours

For more information or to request a Trial of this feature, please contact a Sportradar Sales representative.

The Open API specification and Syntax for using our Probabilities feeds, including examples of the payloads can be found below.

OpenAPI Specification

Live Probabilities

Provides top-level information for live matches. If probabilities are available for a match, pre-match and live probabilities will be displayed.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties - Brackets Competition Properties - Schedules Competition Properties - Season Player Stats Competition Properties - Season Stats Leaders Competition Properties - Season Team Stats Competition Properties - Standings Competition Properties - Team Squads	Coverage Type Group - Group Name Group Id Group Name Round Name Round Number Season Competition Id Season End Date Season Id	Season Name Season Start Date Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score Market Home Score	Market Last Updated Market Live Market Name Market Removed	Match Time Outcome Name Remaining Time Remaining Time in Period

Match Info Data Points:	Away Score Competitor Qualifier Home Score Neutral Ground Period Number Period Type Sport Event Id Sport Event Properties - Basic Play by Play Sport Event Properties - Basic Team Statistics	Sport Event Properties - Deeper Player Stats Sport Event Properties - Deeper Team Stats Sport Event Properties - Lineups Sport Event Properties - Goal Scorers Sport Event Properties - Scores Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time	Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.com/icehockey-probabilities/trial/v2/en/schedules/live/probab
ilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey-probabilities/`{access_level}`/`{version}`/`{language_code}`/schedules/live/probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Season Outright Probabilities

Provides a list of outright probabilities for each competitor from a given season.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Probability Info Data Points:	Win Probability	Market Name	Outcome Name

```
curl -X GET "https://api.sportradar.com/icehockey-probabilities/trial/v2/en/seasons/sr:season:789
17/outright_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Outright Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey-probabilities/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/outright_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Probabilities

Provides pre-match and live probabilities for a given match.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id	Coverage Type Group - Group Name	Season Name Season Start Date

	Competition Name Competition Parent Id Competition Properties - Brackets Competition Properties - Schedules Competition Properties - Season Player Stats Competition Properties - Season Stats Leaders Competition Properties - Season Team Stats Competition Properties - Standings Competition Properties - Team Squads	Group Id Group Name Round Name Round Number Season Competition Id Season End Date Season Id	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score Market Home Score	Market Last Updated Market Live Market Name Market Removed	Match Time Outcome Name Remaining Time Remaining Time in Period
Match Info Data Points:	Away Score Competitor Qualifier Home Score Neutral Ground Period Number Period Type Sport Event Id Sport Event Properties - Basic Play by Play Sport Event Properties - Basic Team Statistics	Sport Event Properties - Deeper Player Stats Sport Event Properties - Deeper Team Stats Sport Event Properties - Lineups Sport Event Properties - Goal Scorers Sport Event Properties - Scores Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time	Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id

```
curl -X GET "https://api.sportradar.com/icehockey-probabilities/trial/v2/en/sport_events/sr:sport_event:24663546/sport_event_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Sport Event Upcoming Probabilities

Provides a list of IDs for upcoming sport events in the next 24 hours.

TTL / Cache:	**1 second**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Match Info Data Points:	Sport Event Upcoming Id

```
curl -X GET "https://api.sportradar.com/icehockey-probabilities/trial/v2/en/sport_events/upcoming
_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Upcoming Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/upcoming_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v2).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Timeline Probabilities

Provides a timeline of pre-match and live probability changes for a given match. This also provides match information including player stats, team stats, and scoring info.

TTL / Cache:	**1 second**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties - Brackets Competition Properties - Schedules Competition Properties - Season Player Stats Competition Properties - Season Stats Leaders Competition Properties - Season Team Stats Competition Properties - Standings Competition Properties - Team Squads	Coverage Type Group - Group Name Group Id Group Name Round Name Round Number Season Competition Id Season End Date Season Id	Season Name Season Start Date Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Abbreviation Age Group Country	Country Code Gender Id	Name
Player Info Data Points:	Id	Name	

Venue Info Data Points:	Capacity Changed City Country Code	Country Name Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max
Probability Info Data Points:	Away Team Win Probability Home Team Win Probability Market Away Score Market Home Score	Market Last Updated Market Live Market Name Market Removed	Match Time Outcome Name Remaining Time Remaining Time in Period
Match Info Data Points:	Away Score Competitor Qualifier Home Score Neutral Ground Period Number Period Type Sport Event Id Sport Event Properties - Basic Play by Play Sport Event Properties - Basic Team Statistics	Sport Event Properties - Deeper Player Stats Sport Event Properties - Deeper Team Stats Sport Event Properties - Lineups Sport Event Properties - Goal Scorers Sport Event Properties - Scores Sport Event Replaced By Sport Event Resumed Time Sport Event Start Time	Sport Event Start Time Confirmed Sport Event Status Sport Event Status Aggregate Home Score Sport Event Status Aggregate Winner Id Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Scout Abandoned Sport Event Status Winner Id
Competitor Match Statistics Data Points:	Goals Goals Conceded Goals in Power Play Goals While Short Handed	Penalties Penalty Minutes Power Plays Puck Possessions	Saves Shots on Target Shutouts Suspensions Minutes Suspensions Number
Player Match Statistics Data Points:	Assists First Assists Goalie Minutes Played	Penalties Penalty Minutes Plus Minus Points	Saves Second Assists Shots on Target

```
curl -X GET "https://api.sportradar.com/icehockey-probabilities/trial/v2/en/sport_events/sr:sport
_event:28274474/timeline_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Timeline Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.com/icehockey-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/timeline_probabilities.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v2).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), English (en), Finnish (fi), French (fr), Russian (ru), Spanish (es), Swedish (sv) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

Return to top

Frequently Asked Questions

Q: What leagues do you cover, and at what level?

Coverage information can be found via our Coverage Matrix.

Q: What format are date fields presented in?

A: When we present date only values we present these in the ISO 8601 standard format.
ex: 2013-04-03
We use these for attributes that have date and no time (such as birthdate). For more information:
https://en.wikipedia.org/wiki/ISO_8601

Q: What format are the date/time fields presented in?

A: All of our Date/Time attributes are in UTC, presented in the ISO 8601 standard format.
ex: 2013-04-03T18:15:00+00:00
For more information: https://en.wikipedia.org/wiki/ISO_8601

Q: How can I find the values for various enum data points within the API?

A: Many enum values are listed in the FAQ below. For the most up-to-date values, please see the Schema section of the OpenAPI specification here: https://api.sportradar.com/icehockey/trial/v2/openapi/swagger/index.html

Q: What are the valid sport event status (sport_event_status – status) values?

A: Here are the valid sport event status values and their definitions:

- not_started – The match is scheduled to be played
- live – The match is currently in progress
- postponed – The match has been postponed to a future date
- delayed – The match has been temporarily delayed and will be continued
- interrupted - The match began, but coverage has stopped for a short time. Note that match scores may not be updated during this period, the last recorded match score will be displayed instead
- cancelled – The match has been cancelled and will not be played
- ended – The match is over

- closed – The match results have been confirmed.

Q: What are the valid match status (sport_event_status – match_status) values?

A: Here are the valid sport event status values and their definitions:

- not_started – The match is scheduled to be played
- live – The match is currently in progress
- 1st_period – The match is in the first period
- 2nd_period – The match is in the second period
- 3rd_period – The match is in the third period
- overtime – The match is in overtime
- 1st_extra – The match is in the first extra period
- 2nd_extra – The match is in the second extra period
- 3rd_extra – The match is in the third extra period
- awaiting_penalties – Waiting for announcement of penalties
- penalties – Penalties are ongoing
- pause – The match is paused
- awaiting_extra_time – Waiting on referee to announce extra time
- interrupted – The match has been interrupted
- abandoned – The match has been abandoned
- postponed – The match has been postponed to a future date
- delayed – The match has been temporarily delayed and will be continued
- cancelled – The match has been cancelled and will not be played
- ended – The match is over
- aet – The match is after extra time
- ap – The match is in after penalties

Q: How do I access past seasons' result and stats?

A: Use the Competition Seasons endpoint to locate the season_id for the season you want to access. Use that season_id to interrogate the various seasons endpoints.

Q: What are the valid outcomes for probabilities?

A: Here are the valid outcome probabilities:

- home_team_winner
- away_team_winner

Q: How do I find out the coverage for a particular match?

A: Within the Summaries, and Timeline feeds, find the node for "coverage". Coverage nodes have two types: season level and sport_event level.
The season level describes data coverage you can expect for matches involved in that given season.
The sport_event level, this exists to describe the data depth of a specific match within the group and season.
Note: There are occasions when the sport_event coverage can vary from the anticipated season coverage and this node exists to highlight that in that event to assist in handling any discrepancies.

Q: Why do different groups have coverage information for a tournament?

A: The notion of “groups” is versatile and is used to distinguish between playoffs, and our competition structures and is therefore necessary to describe coverage at a group level for consistency. Generally, however, within a competition in Ice Hockey, there will be no difference between competition coverage between conferences.

Q: What are the possible event types logged?

A: Here are all of the possible event types we log:

- match_started
- period_started
- period_score
- score_change (goal_scorer, primary_assist, secondary_assist)
- suspension
- break_start
- match_ended

Q: What are markets and what are the different markets?

Markets are something you can bet on, for which we provide probabilities. Over time we intend to provide more and more markets in the API. Currently we provide 3-way (home win, away win, or draw) and Next Goal (which team will score the next goal) markets.

Q: What are the possible outcomes?

A: Different markets have different outcomes. The available markets are currently:

- home_team_winner
- away_team_winner
- draw
- none

Q: Why are the “shots_on_target” incorrect for Finland Liiga?

A: At this time, the league does not provide shots on target instead (for Finland Liiga only) we provide this data point as a total number of shots.

Q: What are the possible player positions in the feeds?

A: Here are the valid player - types:

- F - Forward
- D - Defenseman
- G – Goalie
- F-D – Forward/Defenseman
- LW – Left Wing
- RW – Right Wing
- C - Center

Q: How are group IDs delivered in the stage array with the various types?

A: With the type of "league" they will have a sr:league prefix. With the type of "cup" they will have a sr:cup prefix.

Q: How are "live" feeds handled in the API?

A: Sport Events appear in the feed 10 minutes before the scheduled start time and are removed 10 minutes after the Sport Event is ended.

Q: What is the probabilities package?

A: The probabilities package is an add-on set of feeds that are an extension of the Season Probabilities feed in the main package (which already provides pre-match probabilities for the sport event winner market). The main features of the Probabilities extension are: Live Probabilities that update throughout game, Season Outright Probabilities for the Tournament Winner market, and Live Probabilities Coverage indicator for the next 24 hours.

Q: Is there Live Probability coverage for every game you cover in the Global Ice Hockey API?

A: Live Probability coverage depends on a number of factors and we can't guarantee that every game will be covered for any given league, however, Sportradar aims to cover close to 100% of games from the following Tier 1 league: National Hockey League (NHL) – USA

Q: Can I utilize the Probabilities extension if I license the Sportradar NHL API?

A: If you have a license for our NHL API you can also integrate Live Probabilities by utilizing the ID Mapping feeds available in Global Ice Hockey v2 to connect games between the two products.

Q: What are the possible standings types I can observe in the Standings feed?

A: Listed below are the values for standing – type.

- total
- home
- away
- first_half_total
- first_half_home
- first_half_away
- second_half_total
- second_half_home
- second_half_away

Q: Why can't I find a particular match in the Daily Summaries, Season Summaries or Sport Events Updated feeds?

A: These endpoints support pagination and return 200 entries by default. To return more matches, an additional query string parameter must be used after your API key. For example, appending *&start=200* will return the next 200 entries per page, *&start=400* will return the next 200, and so on.

Q: How will a sport event behave when it is not covered with live scores?

A: When a sport_event is not covered live, the status and match_status will remain as not_started until results are entered post-match.

Q: What are the possible values for break_name in the feeds?

A: Here are the valid break names:

- 1st_pause
- 2nd_pause
- awaiting_extra
- awaiting_penalties

Q: What are the possible values for player type in the feeds?

A: Here are the valid player type values:

- scorer
- assist
- secondary_assist

Q: How do I locate the TTL (Time to Live)/cache on an API endpoint?

A: The cache (in seconds) can be accessed in the returned header information on each RESTful API call, under `cache-control`. ex. `cache-control: max-age=1, public, s-maxage=1` or `cache-control: public, must-revalidate, max-age=120`

Q: What are the possible values for `cup_round` – `state` in the Season Links feed?

A: Listed below are the values and definitions for `cup_round` - `state`. These can be leveraged to determine the status of a cup round.

- `empty` - A matchup has been created but neither the match details nor the competitors are known.
- `unseeded_fixture` - Match details are known but competitors are unknown.
- `partial_seeded` - One competitor is known.
- `partial_seeded_fixture` - Match details and one competitor are known.
- `seeded` - Both competitors are known.
- `seeded_fixture` - Match details and both competitors are known.
- `unstarted` - Match(es) have been added.
- `on_going` - The first match has started.
- `decided` - The last match has ended.
- `winner` - The winner is known.
- `cancelled` – The matchup has been cancelled.

Return to top

SPORTRADAR
150 South 5th St. Suite 400
Minneapolis, MN 55402
USA

us-sales@sportradar.com
us-jobs@sportradar.com
Main: 612-361-4100
Sales: 612-361-4102
Terms and Conditions

SPORTRADAR IS SOCIAL

Privacy Policy © Sportradar, a subsidiary of Sportradar AG

Facebook Technology Partner

Exhibit 57

Soccer v4

Statistics Summary

Documentation

OpenAPI Specification

Change Log

Coverage Matrix

Frequently Asked Questions

Quick Access	
Postman Collection	**XSD Schema**
▶ Run in Postman	Download

Soccer API Overview

Our Soccer API provides real-time game updates and a vast database of supplementary statistics for the largest leagues in the world. Data is collected via Sportradar on-venue scouts and in-house operators.

Over 650 unique competitions are available in one package. See our Coverage Matrix for a detailed breakdown of competitions and data offered.

The Soccer API is consistent in structure, format and behavior with the other General Sport APIs. Primary feeds will return seasons, competitions, team/player data and real-time scores.

Additional feeds provide a host of complementary stats, including:

- Live standings for top leagues
- League leaders
- Win probabilities for every match
- Lineups with formations
- Seasonal statistics
- Season leaders
- Missing/Injured players
- Live ball-spotting data (x,y coordinates for events on the field)
- Head-to-head statistics
- Fun facts

Real-time customers are also offered two delivery Push Feeds to enhance speed.

An extra package containing **extended statistics** is also available. This package includes live extended match player stats for:

- Minutes
- Passing
- Tackles
- Crosses
- Blocks/Saves
- Fouls
- Offsides

An extended Probabilities package is also offered. This add-on includes in-game probability updates and season outrights.

API	API Version
Soccer	v4

Note: Each league group requires a separate API key and authentication is required for all API calls.

Soccer v4 API Map

To best utilize the Soccer v4 API, you will need several parameters to create your API calls. The map below illustrates how you can obtain the parameters you need.

Examples:

To find the fun facts for a given match:

1. Call the daily summaries for the date of the match and find the Sport Event Id for the chosen match
2. Call the Sport Event Fun Facts using the Sport Event Id
3. Locate the Fact Statement

The fun facts for the match are displayed.

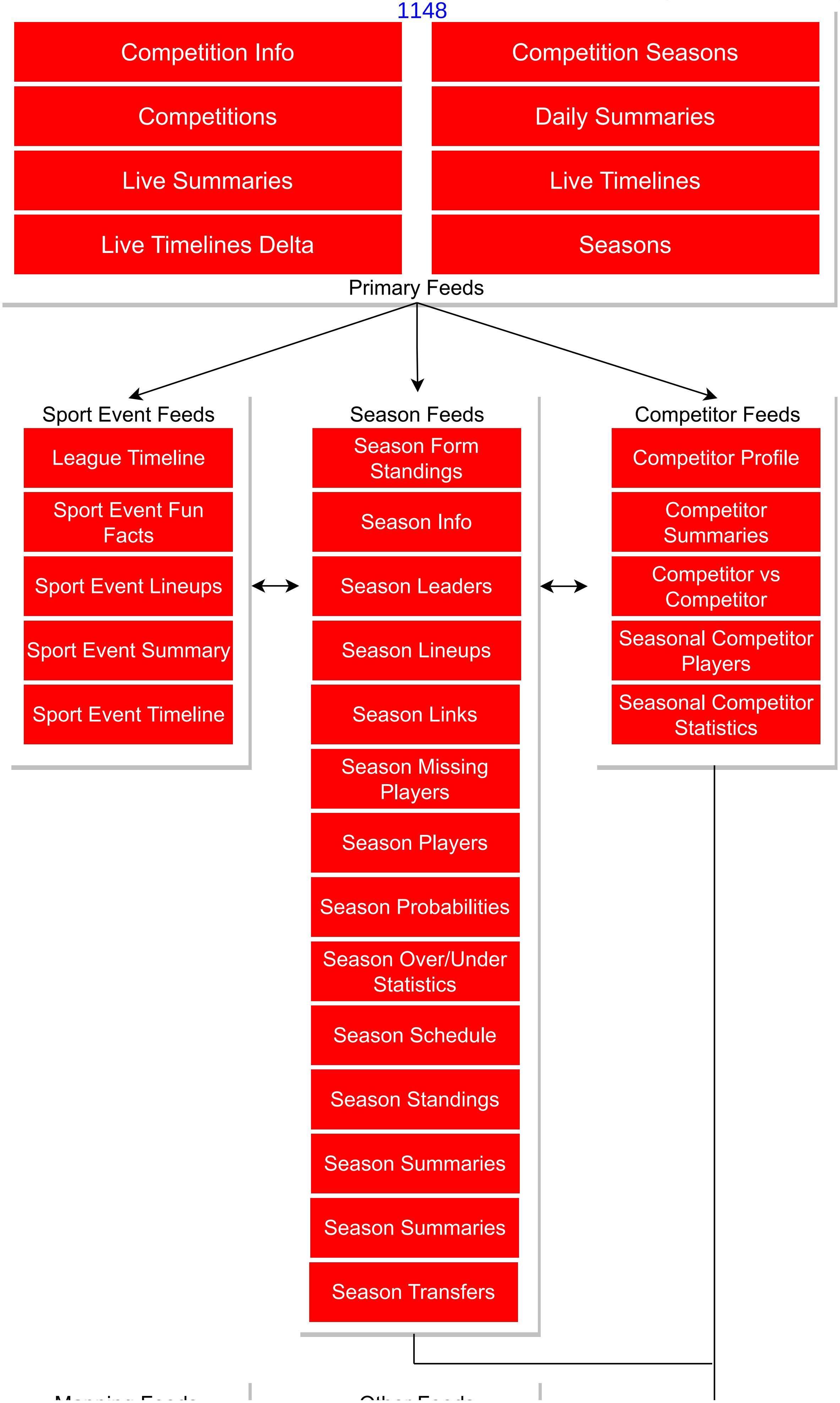
To find a teams number of goals by headers for the current season:

1. Call the daily summaries and find the desired Competitor Id
2. Make note of the Season Id for the current season
3. Call the seasonal competitor statistics using the Team Id and Season Id
4. Locate the statistics - goals_by_head

The teams number of goals by headers is displayed.

Primary and mapping feeds only require dates, human-readable parameters or a Competition Id (which can be found in the competitions endpoint), while other feeds require a global unique identifier (GUID) to identify the sport event, competitor, or player that the feed will describe. The primary feeds provide GUIDs for sport events, competitions, and competitors, while sport event and competitors provide GUIDS for players.

- - Available only with the probabilities plan within the Soccer v4 package. See FAQ for details.



Competition Info

Competition Seasons

Competitions

Daily Summaries

Live Summaries

Live Timelines

Live Timelines Delta

Seasons

Primary Feeds

Sport Event Feeds

- League Timeline
- Sport Event Fun Facts
- Sport Event Lineups
- Sport Event Summary
- Sport Event Timeline

Season Feeds

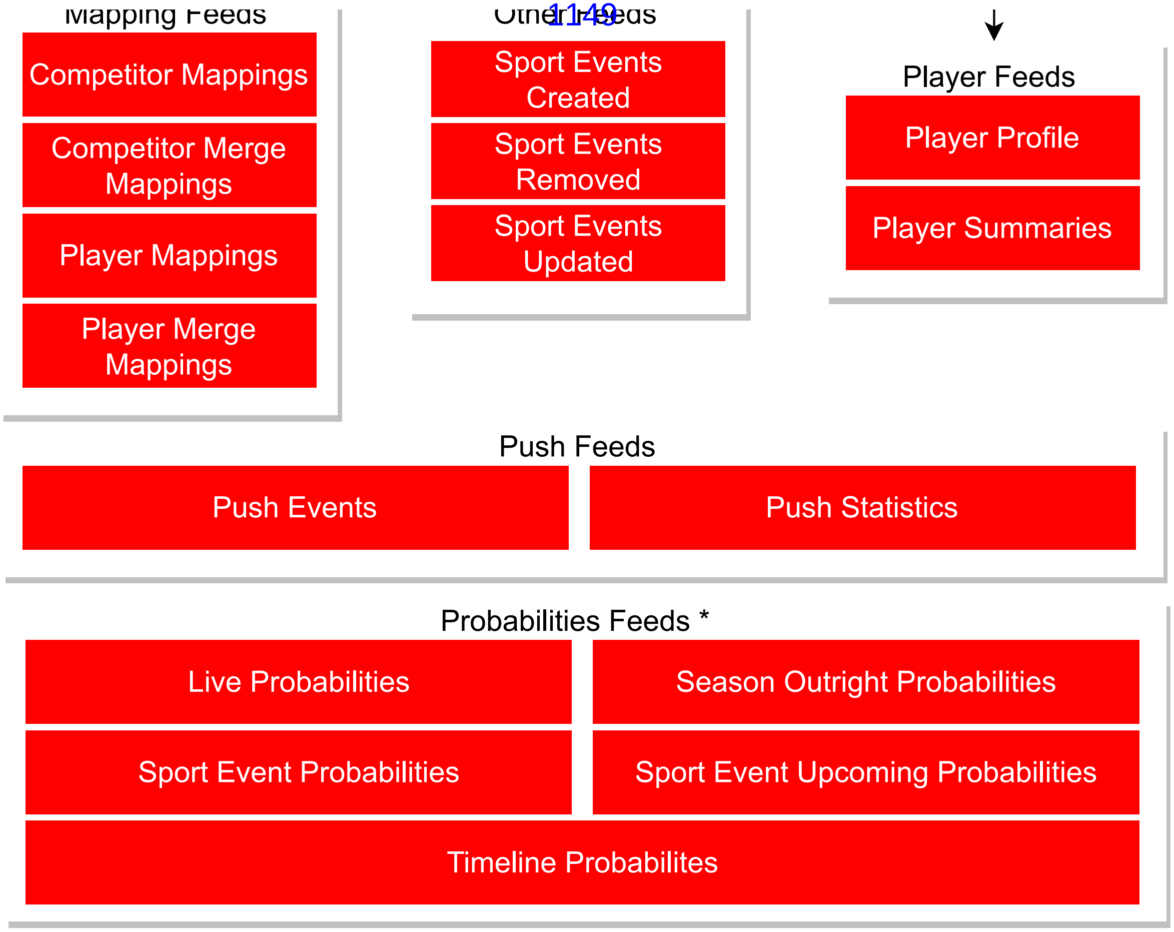
- Season Form Standings
- Season Info
- Season Leaders
- Season Lineups
- Season Links
- Season Missing Players
- Season Players
- Season Probabilities
- Season Over/Under Statistics
- Season Schedule
- Season Standings
- Season Summaries
- Season Summaries
- Season Transfers

Competitor Feeds

- Competitor Profile
- Competitor Summaries
- Competitor vs Competitor
- Seasonal Competitor Players
- Seasonal Competitor Statistics

Mapping Feeds

Other Feeds



Soccer v4 Failover Information

If our scout feed goes down or becomes unavailable, Sportradar takes over Live Data Entry (LDE) to provide a failover.

In case of a failover, we follow one of the following scenarios:

1. We are unable to provide deeper player or team stats, such as shots, corners, assists, and offsides. We will deliver only basic stats and events such as goals, cards, and substitutions with player names.
2. We are unable to provide deeper player or team stats and player names for all basic events, such as cards, substitutions, shots, corners, assists, and offsides. We will display player names for goals.

Competition Info

Provides the name, id, and parent id for a given competition.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Id	Name	Parent Id

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitions/sr:competition:17/info.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Info feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/competitions/`{competition_id}`/info.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competition Info feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitions/getCompetitionInfo

Return to top

Competition Seasons

Provides historical season information for a given competition. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Gender Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitions/sr:competition:17/seasons.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competition Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /competitions/ `{competition_id}` /seasons. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competition_id`	Id of a given competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competition Seasons feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitions/getCompetitionSeasons

Return to top

Competitions

Provides a list of all available Soccer competitions.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Category & Sport Info:	Category Country Code	Category Id	Category Name
Competition Info Data Points:	Competition Gender Competition Id	Competition Name	Competition Parent Id

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitions.xml?api_key={your_api_ke
y}"
```

The above command returns xml structured like this.

Access the Competitions feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/competitions.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitions feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitions/getCompetitions

Return to top

Competitor Mappings

Provides competitor id mapping between previous versions of the Soccer API.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 competitor mappings.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	External Id Id	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitors/mappings.xml?api_key={your
_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /competitors/mappings. `{format}` ?
api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitor Mappings feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitors/getCompetitorMappings

Return to top

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/players/sr:player:159665/summaries.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description

`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=1000

Return to top

Competitor Merge Mappings

Provides the valid Sportradar Id in cases when two competitors have been merged into one. Entries are retained in this endpoint for one week.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	Name Id	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitors/merge_mappings.xml?api_key
={your_api_key}"
```

Access the Competitor Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /competitors/merge_mappings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitor Merge Mappings feed, use the following URL.

https://api.sportradar.com/soccer-extended/trial/v4/openapi/swagger/index.html#/mappings/getCompetitorMergeMappings

Return to top

Competitor Profile

Provides top-level information for a given team, including the full team roster, manager, home venue, and team colors.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name Competitor Qualifier Competitor Virtual Jersey Base Color Jersey Horizontal Stripes	Jersey Horizontal Stripes Color Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Sleeve Detail Jersey Split Jersey Split Color Jersey Squares Jersey Squares Color Jersey Stripes	Jersey Stripes Color Jersey Type Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name Manager Nationality Manager Nickname Manager Preferred Foot
Player Info Data Points:	Country Code Date of Birth Gender Height	Id Jersey Number Name Nationality	Nickname Place of Birth Preferred Foot Type Weight
Venue Info Data Points:	Capacity City Country	Country Code Id	Map Coordinates Name

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitors/sr:competitor:44/profile.x
ml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Profile feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitor Profile feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitors/getCompetitorProfile

Return to top

Competitor Summaries

Provides previous and upcoming match information for a given competitor, including statistics for past matches and scheduling info for upcoming matches.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

		Season End Date Season Id	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Player Match Statistics Data Points:	Assists Corner Kicks Goals Scored	Penalties Missed Red Cards Shots Blocked	Substituted In Substituted Out

	Offsides Own Goals	Shots Off Target Shots On Target	Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitors/sr:competitor:44/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitor Summaries feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitors/getCompetitorSummaries

Return to top

Competitor vs Competitor

Provides previous and upcoming matches between two teams including scoring information, player and team match statistics.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups

	Group Properties Standings Lineups Confirmed	Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Player Match Statistics Data Points:	Assists Corner Kicks Goals Scored Offsides Own Goals	Penalties Missed Red Cards Shots Blocked Shots Off Target Shots On Target	Substituted In Substituted Out Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/competitors/sr:competitor:44/versus/s
r:competitor:42/summaries.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Competitor vs Competitor feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /competitors/ `{competitor_id}` /versus/ `{competitor_id2}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`competitor_id`	Id of a given competitor.
`competitor_id2`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitor vs Competitor feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/summaries/getCompetitorVersusSummaries

Return to top

Daily Summaries

Provides match information for a given day including team scoring, player and team match statistics.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date

	Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id

	Match Situation - Qualifier Match Situation - Status	Normaltime Score Sport Event Status Home Overtime Score	
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Player Match Statistics Data Points:	Assists Corner Kicks Goals Scored Offsides Own Goals	Penalties Missed Red Cards Shots Blocked Shots Off Target Shots On Target	Substituted In Substituted Out Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/schedules/2019-07-23/summaries.xml?api
_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Daily Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /schedules/ `{year}` - `{month}` - `{day}` /summaries. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`year`	Year in 4 digit format (YYYY).
`month`	Month in 2 digit format (MM).
`day`	Day in 2 digit format (DD).

`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Daily Summaries feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/summaries/getScheduleSummaries

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/schedules/2019-07-23/summaries.xml?api_key={you
r_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

League Timeline

Provides an alternate set of statistics for an event which match official league sites.

Official stats are provided for: England Premier League, Germany Bundesliga, Italy Serie A, Spain La Liga, UEFA Champions League, USA MLS, Austria Bundesliga.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name	Competition Properties Season Team Statistics Competition Properties	Season Name Season Start Date Season Competition Id

	Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id Name	Starter	Type
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced	Match Situation - Updated At Sport Event Status Away Normaltime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie

	Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Away Team Score Ball Location Qualifier Ball Location Order Ball Location X Coordinate Ball Location Y Coordinate Event Break Name Event Card Description Event Commentary Event Competitor Event Coordinate X	Event Coordinate Y Event Id Event Injury Time Announced Event Match Clock Event Match Time Event Method Event Outcome Event Period	Event Period Name Event Period Type Event Shootout Away Score Event Shootout Home Score Event Stoppage Time Event Type Event Time Home Team Score
Play Details Data Points:	Assisting Player Id Assisting Player Method Assisting Player Name Assisting Player Type Goal Scorer Id Goal Scorer Method	Goal Scorer Name Goal Scorer Type Substitution Player In Id Substitution Player In Method Substitution Player In Name	Substitution Player In Type Substitution Player Out Id Substitution Player Out Method Substitution Player Out Name Substitution Player Out Type
Player Match Statistics Data Points:	Assists Goals Red Cards	Substituted In Substituted Out	Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Assists Goals	Red Cards Substitutions	Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/sr:sport_event:18427304/l
eague_timeline.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the League Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /league_timeline. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the League Timeline feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventTimeline

Return to top

Live Summaries

Provides match information for all currently live matches including team scoring, player and team match statistics. This feed updates in real time as matches are played. Matches appear a few minutes before kick-off and disappear a few minutes after the match reaches “ended” status.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

	Competition Properties Season Stats Leaders	Round Other Sport Event Id Season End Date Season Id	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Player Match Statistics Data Points:	Assists Corner Kicks	Penalties Missed Red Cards	Substituted In Substituted Out

	Goals Scored Offsides Own Goals	Shots Blocked Shots Off Target Shots On Target	Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/schedules/live/summaries.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Live Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/schedules/live/summaries.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Live Summaries feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/summaries/getScheduleLiveSummaries

Return to top

Live Timelines

Provides a play-by-play event timeline for currently live matches. Matches appear a few minutes before kick-off and disappear a few minutes after the match reaches “ended” status.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Player Info Data Points:	Id	Name	Starter
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Away Team Score Ball Location Qualifier Ball Location Order Ball Location X Coordinate Ball Location Y Coordinate Event Break Name Event Card Description Event Commentary Event Competitor Event Coordinate X Event Coordinate Y	Event Id Event Injury Time Announced Event Match Clock Event Match Time Event Method Event Outcome Event Period Name Event Period Type	Event Shootout Away Score Event Shootout Home Score Event Stoppage Time Event Type Event Time Home Team Score Sport Event Timeline Id Sport Event Timeline Start Time
Play Details Data Points:	Assisting Player Id Assisting Player Method Assisting Player Name Assisting Player Type Goal Scorer Id Goal Scorer Method	Goal Scorer Name Goal Scorer Type Substitution Player In Id Substitution Player In Method Substitution Player In Name	Substitution Player In Type Substitution Player Out Id Substitution Player Out Method Substitution Player Out Name Substitution Player Out Type

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/schedules/live/timelines.xml?api_key=
{your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Live Timelines feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/timelines/getLiveTimelines

Return to top

Live Timelines Delta

Provides a 10 second live delta of match information, including scoring and a play-by-play event timeline.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Player Info Data Points:	Id	Name	Starter
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Away Team Score Ball Location Qualifier Ball Location Order Ball Location X Coordinate Ball Location Y Coordinate	Event Id Event Injury Time Announced Event Match Clock Event Match Time Event Method	Event Shootout Away Score Event Shootout Home Score Event Stoppage Time Event Type Event Time

	Event Break Name Event Card Description Event Commentary Event Competitor Event Coordinate X Event Coordinate Y	Event Outcome Event Period Name Event Period Type	Home Team Score Sport Event Timeline Delta Id Sport Event Timeline Delta Start Time
Play Details Data Points:	Assisting Player Id Assisting Player Method Assisting Player Name Assisting Player Type Goal Scorer Id Goal Scorer Method	Goal Scorer Name Goal Scorer Type Substitution Player In Id Substitution Player In Method Substitution Player In Name	Substitution Player In Type Substitution Player Out Id Substitution Player Out Method Substitution Player Out Name Substitution Player Out Type

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/schedules/live/timelines_delta.xml?api
_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Live Timlines Delta feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/schedules/live/timelines_delta.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Live Timelines Delta feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/timelines/getLiveTimelinesDelta

Return to top

Player Mappings

Provides player id mapping between previous versions of the Soccer API.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1,000 player mappings.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	External Id Id	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/players/mappings.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/players/mappings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Player Mappings feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/players/getPlayerMappings

Return to top

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/players/sr:player:159665/summaries.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=1000

Return to top

Player Merge Mappings

Provides valid ids for players who have had their profiles merged. While Sportradar always strives to provide one unique player id, it is a possibility for two ids to be created. This feed provides the correct id once profiles have been merged. Entries are retained in this endpoint for one week.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Mappings Info Data Points:	Name Id	Merged Id	Retained Id

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/players/merge_mappings.xml?api_key={yo
ur_api_key}"
```

The above command returns xml structured like this.

Access the Player Merge Mappings feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /players/merge_mappings. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Player Merge Mappings feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/players/getPlayerMergeMappings

Return to top

Player Profile

Provides player information, including current and historical team membership info.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Country Code Date of Birth Gender Height Id Jersey Number	Name Nationality Nickname Place of Birth Preferred Foot Team Membership Active Flag	Team Membership End Date Team Membership Jersey Number Team Membership Start Date Team Membership Type Type Weight

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/players/sr:player:159665/profile.xml?a
pi_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Profile feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /players/ `{player_id}` /profile. `{format}` ?api_key= `{your_api_key}`

Parameter	Description

`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`player_id`	Id of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Player Profile feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/players/getPlayerProfile

Return to top

Player Summaries

Provides match info and statistics for the past 10 matches in which a given player participated.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

	Competition Properties Season Stats Leaders	Round Other Sport Event Id Season End Date Season Id	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Player Match Statistics Data Points:	Assists Corner Kicks	Penalties Missed Red Cards	Substituted In Substituted Out

	Goals Scored Offsides Own Goals	Shots Blocked Shots Off Target Shots On Target	Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/players/sr:player:159665/summaries.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Player Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/players/`{player_id}`/summaries.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`player_id`	Id of a given player.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Player Summaries feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/players/getPlayerSummaries

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/players/sr:player:159665/summaries.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Season Competitors

Provides a list of teams participating for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Id	Name Short Name	Competitor Virtual

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/competitors.xm
l?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Competitors feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/competitors.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).

`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Competitors feed, use the following URL.

https://api.sportradar.com/soccer/production/v4/openapi/swagger/index.html#/seasons/getSeasonCompetitors

Return to top

Season Form Standings

Provides a form table of game results and splits for a given season. Table displays W/D/L (win/draw/loss) for a maximum of 6 matches for each team.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Group Id	Group Name	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Standings Data Points:	Draws Form Standing Results Form Standing Type Goals Against	Goal Difference Goals For Losses Played	Points Rank Wins

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:93741/form_standing
s.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Form Standings feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /form_standings. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:93741/form_standings.xml?api
_key={your_api_key}&limit=3'
```

In addition to the URL parameters listed above, you can filter the Standings information with the following optional query string parameter.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

Parameter	Description
`round`	Round number expressed as: {round}. Example: round=1
`limit`	Number of matches played to display for each team. Example: limit=2 *Note: The max limit is 10, which will show for example `played=10`, `win=8`, `loss=2`. But the `form` will always display a max of 6 matches.*

Return to top

Season Info

Provides detailed information for a given season, including participating teams and coverage level.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders Competition Properties Season Team Statistics	Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Max Rounds Group Name Group Parent Group Id Info Competition Status Info Venue Changed Info Venue Reduced Capacity Info Venue Reduced Capacity Max Season End Date	Season Id Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id Competitor Other Season Id	Competitor Name Competitor Qualifier Competitor Virtual
Sport Event Info Data Points:	Coverage Type Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules	Group Properties Standings Sport Event Properties Assists Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats	Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/info.xml?api_k
ey={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Info feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/info.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Info feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonInfo

Return to top

Season Leaders

Provides a list of leaders for a given season. Statistics include points, goals, assists, cards, and minutes played.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation	Competitor Id	Competitor Name
Player Info Data Points:	Player Id	Player Multiple Competitors	Player Name
Season Leader Info Data Points:	List Type Leader Rank	Datapoint Type	Datapoint Value

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/leaders.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Leaders feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/leaders.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Leaders feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonLeaders

Return to top

Season Lineups

Provides match lineups and substitutions for a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**30 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name

Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name Competitor Qualifier Competitor Virtual Jersey Base Color Jersey Horizontal Stripes	Jersey Horizontal Stripes Color Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Sleeve Detail Jersey Split Jersey Split Color Jersey Squares Jersey Squares Color Jersey Stripes	Jersey Stripes Color Jersey Type Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name Manager Nationality Manager Nickname Manager Preferred Foot
Player Info Data Points:	Country Code Date of Birth Gender Height Id	Jersey Number Name Nationality Nickname Place of Birth	Played Preferred Foot Starter Type Weight
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock

	Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	Ball Location X Coordinate	Ball Location Y Coordinate
Match Lineups Data Points:	Player Order Player Position	Team Designation	Team Formation
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/lineups.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Lineups feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /lineups. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages

	Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Lineups feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonLineups

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/seasons/sr:season:66441/lineups.xml?api_key={yo
ur_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, suggested maximum value is 75. Example: limit=75

Return to top

Season Links

Provides information about linked cup rounds for a given season.

Use this feed to compile full advancement brackets for relevant seasons/tournaments. Links between all matches and rounds are available when competitors (TBD vs. TBD) are not yet known.

TTL / Cache:	**300 seconds**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Competition Info Data Points:	Cup Round Id Cup Round Name Cup Round Order Cup Round State Cup Round Type Cup Round Winner Id	Group - Group Name Group Id Stage End Date Stage Order Stage Phase	Stage Start Date Stage Type Stage Year
Sport Event Info Data Points:	Sport Event Id	Sport Event Start Time	Sport Event Start Time Confirmed

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:74551/stages_groups_
cup_rounds.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Links feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /stages_groups_cup_rounds. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Links feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonStagesGroupsCupRounds

Return to top

Season Missing Players

Provides a list of injured and/or missing players for a given season.

TTL / Cache:	**300 seconds**

Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation	Competitor Id	Competitor Name
Player Info Data Points:	Id Missing Reason	Missing Start Date Name	Status

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/missing_player
s.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Missing Players feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /missing_players. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Missing Players feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonMissingPlayers

Return to top

Season Over/Under Statistics

Provides the over/under match goal totals for all teams in a given season.

TTL / Cache:	**300 seconds**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Id	Competitor Name	
Team Season Statistics Data Points:	Over/Under List Type Over/Under - Goals	Over/Under - Matches Over	Over/Under - Matches Under

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:83926/over_under_statistics.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Over/Under Statistics feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /over_under_statistics. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Over/Under Statistics feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonOverUnderStatistics

Return to top

Season Players

Provides names and ids for all participating players for a given season.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 1000 players.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Player Info Data Points:	Country Code Date of Birth Display First Name Display Last Name First Name	Gender Height Id Jersey Number Last Name	Name Nationality Nickname Type Weight

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/players.xml?ap
i_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Players feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /players. `{format}` ?api_key= `{your_api_key}`

Parameter	**Description**
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Players feed, use the following URL.

https://api.sportradar.com/soccer/production/v4/openapi/swagger/index.html#/seasons/getSeasonPlayers

Return to top

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/seasons/sr:season:66441/players.xml?api_key={yo
ur_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This may be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=1000

Return to top

Season Probabilities

Provides 3-way win probabilities (home team win, away team win, draw) for all matches for a given season.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

		Season End Date Season Id	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Sport Event Info Data Points:	Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Lineups Confirmed	Group Properties Standings Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats	Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Match Probabilities Data Points:	Away Team Win Probabilities Draw Probabilities	Home Team Win Probabilities Market Name	Outcome Name
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/probabilities.
```

```
xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Probabilities feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /seasons/ `{season_id}` /probabilities. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Probabilities feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonProbabilities

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/probabilities.xml?api_
key={your_api_key}&start=200'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0

`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Season Schedule

Provides basic match information for all matches for a given season, including scoring and match coverage.

Note: Pagination may be required to pull all data within this feed. By default, the feed will return 1000 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Sport Event Info Data Points:	Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League	Group Properties Standings Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time	Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats

	Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules	Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats	Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/schedules.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Schedule feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/schedules.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi),

	French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Schedule feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonSchedules

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/seasons/sr:season:66441/schedule.xml?api_key={your_api_key}&start=0&limit=100'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 1000. Example: limit=1000

Return to top

Season Standings

Provides detailed standings info for a given season.

TTL / Cache:	**10 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Group - Group Name Group Id	Stage End Date Stage Order	Stage Start Date Stage Type

	Group Name Group Parent Group Id	Stage Phase	Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Form Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Standings Info Data Points:	Change Current Outcome Draws Goals Against Goals Differential Goals For Group - Group Name Group Comment Text	Group Id Group Live Group Name Losses Matches Played Points Points Draw Points Loss	Points Per Game Points Win Rank Round Tie Break Rule Type Wins

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/standings.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Standings feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/standings.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/standings.xml?api_key=
{your_api_key}&round=1'
```

In addition to the URL parameters listed above, you can filter the Standings information with the following optional query string parameter.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

Parameter	Description
`round`	Round number expressed as: {round}. Example: round=1
`live`	Live standings expressed as a boolean value. Example: live=true

To retrieve the OpenAPI Schema Definition for the Season Standings feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonStandings

Return to top

Season Summaries

Provides information for all matches from a given season including scoring and statistics at the match level.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 100 sport events.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

	Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate

Player Match Statistics Data Points:	Assists Corner Kicks Goals Scored Offsides Own Goals	Penalties Missed Red Cards Shots Blocked Shots Off Target Shots On Target	Substituted In Substituted Out Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/summaries.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Summaries feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/summaries.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Summaries feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonSummaries

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/seasons/sr:season:66441/summaries.xml?api_key=
{your_api_key}&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 100. Example: limit=100

Return to top

Season Transfers

Provides a list of all player transfers for a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Country Code Date of Birth Gender Height Id Jersey Number	Name Nationality Nickname Place of Birth Transfer Date	Transfer From Competitor Transfer Role Type Transfer To Competitor Type Weight

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:83706/transfers.xml?
api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Season Transfers feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/transfers.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Summaries feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasonTransfers

Return to top

Seasonal Competitor Players

Provides player roster information for every team from a given season.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competitor Info Data Points:	Abbreviation Id	Name	Short Name
Player Info Data Points:	Country Code Date of Birth Gender Height Id	Jersey Number Name Nationality Nickname	Place of Birth Preferred Foot Type Weight

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:66441/competitor_players.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasonal Competitor Players feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/competitor_players.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Seasonal Competitor Players feed, use the following URL.

https://api.sportradar.com/soccer/production/v4/openapi/swagger/index.html#/seasons/getSeasonCompetitorPlayers

Return to top

Seasonal Competitor Statistics

Provides team and player seasonal statistics for a given season.

TTL / Cache:	**30 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Id Competition Name Competition Parent Id	Season Competition Id Season End Date Season Id	Season Name Season Start Date Season Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter

Player Seasonal Statistics Data Points:	Assists Corner Kicks Goals Scored Offsides Own Goals	Penalties Missed Red Cards Shots Blocked Shots Off Target Shots On Target	Substituted In Substituted Out Yellow Cards Yellow/Red Cards
Team Seasonal Statistics Data Points:	Average Ball Possession Cards Given Corner Kicks Free Kicks Goals by Foot Goals by Head Goals Conceded Goals Conceded First Half	Goals Conceded Second Half Goals Scored Goals Scored First Half Goals Scored Second Half Offsides Penalties Missed Red Cards Shots Blocked	Shots off Target Shots on Bar Shots on Post Shots on Target Shots Total Yellow Cards Yellow/Red Cards

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons/sr:season:54571/competitors/sr:competitor:17/statistics.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasonal Competitor Statistics feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/competitors/`{competitor_id}`/statistics.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`competitor_id`	Id of a given competitor.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Competitor Seasonal Statistics feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/competitors/getSeasonCompetitorStatistics

Return to top

Seasons

Provides a list of historical season information for all competitions. Competitions will return a maximum of three seasons of data, including current or newly created seasons.

TTL / Cache:	**300 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Competition Info Data Points:	Season Competition Id Season End Date	Season Id Season Name	Season Start Date Season Year

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/seasons.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Seasons feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/seasons.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Seasons feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/seasons/getSeasons

Return to top

Sport Event Fun Facts

Provides noteworthy, human-readable, facts based on statistical information about a given match and its competing teams.

TTL / Cache:	**300 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON
Sport Event Info Data Points:	Match Facts

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/sr:sport_event:18427304/f
un_facts.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Fun Facts feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /fun_facts. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Event Fun Facts feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventFunFacts

Return to top

Sport Event Lineups

Provides detailed roster information for a given match. Starting players, substitutions, formation type, and channel availability are included if supported by coverage level.

TTL / Cache:	**1 second**
Update Frequency:	As Necessary

Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country Competitor Country Code Competitor Gender Competitor Id Competitor Name Competitor Qualifier Competitor Virtual Jersey Base Color Jersey Horizontal Stripes	Jersey Horizontal Stripes Color Jersey Number Color Jersey Shirt Type Jersey Sleeve Color Jersey Sleeve Detail Jersey Split Jersey Split Color Jersey Squares Jersey Squares Color Jersey Stripes	Jersey Stripes Color Jersey Type Manager Country Code Manager Date of Birth Manager Gender Manager Id Manager Name Manager Nationality Manager Nickname Manager Preferred Foot
Player Info Data Points:	Country Code Date of Birth Gender Height Id	Jersey Number Name Nationality Nickname Place of Birth	Played Preferred Foot Starter Type Weight
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats

	Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	Ball Location X Coordinate	Ball Location Y Coordinate
Match Lineups Data Points:	Player Order Player Position	Team Designation	Team Formation
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/sr:sport_event:18427304/l
ineups.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Lineups feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /lineups. `{format}` ?api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Event Lineups feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventLineups

Return to top

Sport Event Summary

Provides real-time match-level statistics for a given match. Including player and team stats, scoring info, and channel availability. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name

Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year
Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions

Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Qualifier Order	X Coordinate	Y Coordinate
Player Match Statistics Data Points:	Assists Corner Kicks Goals Scored Offsides Own Goals	Penalties Missed Red Cards Shots Blocked Shots Off Target Shots On Target	Substituted In Substituted Out Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/sr:sport_event:18427304/s
ummary.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Summary feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/summary.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi),

	French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Event Summary feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventSummary

Return to top

Sport Event Timeline

Provides real-time match-level statistics and a play-by-play event timeline for a given match. This includes player and team stats, scoring info, channel availability, x/y event coordinates, and human-readable event descriptions. Please note that data returned is determined by coverage level.

TTL / Cache:	**1 second**		
Update Frequency:	Realtime		
Content Type:	XML or JSON		
Sport & Category Info Data Points:	Category Country Code Category Id	Category Name Sport Id	Sport Name
Competition Info Data Points:	Competition Gender Competition Id Competition Name Competition Parent Id Competition Properties Brackets Competition Properties Missing Players Competition Properties Player Transfer History Competition Properties Schedules Competition Properties Season Player Statistics Competition Properties Season Probabilities Competition Properties Season Stats Leaders	Competition Properties Season Team Statistics Competition Properties Standings Competition Properties Team Squads Group Group Name Group Id Group Name Round Competition Sport Event Number Round Cup Round Number of Sport Events Round Cup Round Sport Event Number Round Name Round Number Round Other Sport Event Id Season End Date Season Id	Season Name Season Start Date Season Competition Id Season Year Stage End Date Stage Order Stage Phase Stage Start Date Stage Type Stage Year

Competitor Info Data Points:	Competitor Abbreviation Competitor Age Group Competitor Country	Competitor Country Code Competitor Gender Competitor Id	Competitor Name Competitor Qualifier Competitor Virtual
Player Info Data Points:	Id	Name	Starter
Sport Event Info Data Points:	Attendance Count Channel Country Channel Country Code Channel Name Channel URL Coverage Type Ground Neutral Group Properties Brackets Group Properties Cup Group Properties Group Stage Group Properties League Group Properties Missing Players Group Properties Qualification Group Properties Results Group Properties Schedules Group Properties Standings Lineups Confirmed	Referee Id Referee Name Referee Nationality Referee Country Code Referee Type Sport Event Id Sport Event Replaced By Sport Event Resume Time Sport Event Start Time Sport Event Start Time Confirmed Sport Event Properties Ballspotting Sport Event Properties Basic Play-by-Play Sport Event Properties Basic Player Stats Sport Event Properties Basic Team Stats Sport Event Properties Commentary	Sport Event Properties Deeper Play-by-Play Sport Event Properties Deeper Player Stats Sport Event Properties Deeper Team Stats Sport Event Properties Extended Player Stats Sport Event Properties Extended Team Stats Sport Event Properties Fun Facts Sport Event Properties Game Clock Sport Event Properties Goal Scorers Sport Event Properties Lineups Sport Event Properties Lineups Availability Sport Event Properties Probabilities Sport Event Properties Scores Sport Event Properties Venue Weather Overall Conditions Weather Pitch Conditions
Boxscore Info Data Points:	Clock Played Clock Stoppage Time Announced Clock Stoppage Time Played Period Score Away Score Period Score Home Score Period Score Number Period Score Type Match Situation - Qualifier Match Situation - Status	Match Situation - Updated At Sport Event Status Away Normaltime Score Sport Event Status Away Overtime Score Sport Event Status Away Score Sport Event Status Decided by Fed Sport Event Status Home Normaltime Score Sport Event Status Home Overtime Score	Sport Event Status Home Score Sport Event Status Match Status Sport Event Status Match Tie Sport Event Status Scout Abandoned Sport Event Status Status Sport Event Status Winner Id
Play by Play Info Data Points:	Away Team Score Ball Location Qualifier Ball Location Order Ball Location X Coordinate Ball Location Y Coordinate Event Break Name Event Card Description Event Commentary	Event Coordinate Y Event Id Event Injury Time Announced Event Match Clock Event Match Time Event Method Event Outcome Event Period	Event Period Name Event Period Type Event Shootout Away Score Event Shootout Home Score Event Stoppage Time Event Type Event Time Home Team Score

	Event Competitor Event Coordinate X		
Play Details Data Points:	Assisting Player Id Assisting Player Method Assisting Player Name Assisting Player Type Goal Scorer Id Goal Scorer Method	Goal Scorer Name Goal Scorer Type Substitution Player In Id Substitution Player In Method Substitution Player In Name	Substitution Player In Type Substitution Player Out Id Substitution Player Out Method Substitution Player Out Name Substitution Player Out Type
Player Match Statistics Data Points:	Assists Corner Kicks Goals Scored Offsides Own Goals	Penalties Missed Red Cards Shots Blocked Shots Off Target Shots On Target	Substituted In Substituted Out Yellow Cards Yellow/Red Cards
Team Match Statistics Data Points:	Ball Possession Cards Given Corner Kicks Fouls Free Kicks Goal Kicks	Injuries Offsides Penalties Missed Red Cards Shots Blocked Shots Off Target	Shots On Target Shots Saved Shots Total Substitutions Throw Ins Yellow Cards Yellow/Red Cards
Venue Info Data Points:	Capacity Changed City Country	Country Code Id Map Coordinates	Name Reduced Capacity Reduced Capacity Max

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/sr:sport_event:18427304/t
imeline.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Event Timeline feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/timeline.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.

`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Event Timeline feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventTimeline

Return to top

Sport Events Created

Provides ids for sport events that have been created in the last 24 hours.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**60 seconds**		
Update Frequency:	As Necessary		
Content Type:	XML or JSON		
Sport Event Info Data Points:	Active Season Flag	Id	Created At

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/created.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Created feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /sport_events/created. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.

`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Events Created feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventsCreated

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/sport_events/created.xml?api_key={your_api_key}
&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Sport Events Removed

Provides ids for sport events that have been removed from the API due to an entry error. Ids will remain in the response for 2 weeks.

Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.

TTL / Cache:	**60 seconds**
Update Frequency:	As Necessary
Content Type:	XML or JSON

Sport Event Info Data Points:	Sport Event Removed Id	Sport Event Removed Replaced By

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/removed.xml?api_key={your
_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Removed feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/`{language_code}`/sport_events/removed.`{format}`?api_key=`{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Events Removed feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventsRemoved

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/sport_events/removed.xml?api_key={your_api_key}
&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Sport Events Updated

Provides ids for sport events that have been updated in the last 24 hours.

> **Note: Pagination will often be required to pull all data within this feed. By default, the feed will return 200 sport events.**

TTL / Cache:	**60 seconds**	
Update Frequency:	As Necessary	
Content Type:	XML or JSON	
Sport Event Info Data Points:	Sport Event Updated Id	Sport Event Updated - Updated At

```
curl -X GET "https://api.sportradar.com/soccer/trial/v4/en/sport_events/updated.xml?api_key={your
_api_key}"
```

The above command returns xml structured like this.

Access the Sport Events Updated feed by replacing the parameters in the following URL:

https://api.sportradar.us/soccer/ `{access_level}` / `{version}` / `{language_code}` /sport_events/updated. `{format}` ? api_key= `{your_api_key}`

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.

`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Events Updated feed, use the following URL.

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventsUpdated

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.us/soccer/trial/v4/en/sport_events/updated.xml?api_key={your_api_key}
&start=0&limit=75'
```

In addition to the URL parameters listed above, you can paginate the response with one or more of the following optional query string parameters. This will often be required to access all available data in this feed.

Note: Optional query string parameters must be added after your API key with an ampersand (&).

Replace placeholders with the following query parameters:

Parameter	Description
`start`	Number to start the list of results from. Example: start=0
`limit`	Number to limit the number of results. Minimum value is 1, maximum value is 200. Example: limit=200

Return to top

Push Feeds

To best utilize Push feeds, we have included code samples in Ruby, Java and Python which provide an example of a way you can consume the feeds. Using these samples will output the feeds content to STDOUT.
For Java, we have also provided a Stream Client to assist your integration.

Note: In the provided Java sample, replace "URL GOES HERE" with the desired Push feed URL.

```
package com.sportradar.http.stream.client;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class StreamClientTest {

    private StreamClient client;
    private static String SERVICE_URL = "<url goes="" here="">";

    @Before
    public void setup() {
        client = new StreamClient();
    }

    @After
    public void cleanup() {
        client.terminate();
    }

    @Test
    public void testStream() throws Exception {
        Handler handler = new ConsoleHandler();
        client.stream(SERVICE_URL, handler);
        System.out.println("Connecting....");
        Thread.sleep(1 * 60 * 1000);
        System.out.println("Disconnecting....");
    }

}
```

Some of our APIs include Push feeds that allow you to get updates as soon as they are available. Push API feeds automatically send JSON and XML payloads to you via a push service, and can dramatically reduce the number of calls you need to make to our RESTful API feeds. The structure of the Push feeds are similar to the structure of the corresponding RESTful API feed (i.e. Push Events and Push Statistics). The push service ensures reliable and efficient delivery of the most up to date information.

Our Push services are based on a HTTP publish/subscribe model. When making a call to the Push APIs, you "subscribe" to various data feeds provided by our service; whenever new content is available on one of those feeds, the server pushes that information out to your client. When no new information is available on the feed, a heartbeat message is sent every 5 seconds to keep the connection active. If you want to filter the results of the feeds, there are several optional query string parameters that can be applied to the API call. If left unfiltered, than all data for the feed is displayed (i.e. all games, events, or statistics).

For your applications to accept data from our Push feeds, ensure that your application can:

- Can follow a HTTP redirect or use the location provided in the feeds header within one minute of your initial request.
- Can accept HTTP data transfer encoded as chunked.

Our Push service does not provide a "stateful session", there is no memory of what data has been sent previously. If you are disconnected from the Push session, you can use the RESTful API to catch up or recover from the disconnection.

Syntax for using our Push feeds and examples of the JSON and XML payloads can be found below.

Push Events

```
curl -L GET 'api.sportradar.com/soccer/trial/v4/stream/events/subscribe?api_key={your_api_key}'
```

The above command returns json like this.

Provides real-time event updates for all live matches.

https://api.sportradar.com/soccer/`{access_level}`/`{version}`/stream/events/subscribe?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'https://api.sportradar.com/soccer/trial/v4/stream/events/subscribe?api_key={your_api
_key}&amp;format=json&amp;sport_event_id=sr:match:13468929'
```

In addition to the URL parameters listed above, you can filter the Events information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

URL Parameters	Description
`competition_id`	Competition id expressed as: {competition_id}. Example: competition_id=sr:competition:204
`event_id`	Event type expressed as: {event_type}. Example: event_id=free_kick
`format`	Format type expressed as: {format}. Example: format=json
`season_id`	Season id expressed as: {season_id}. Example: season_id=sr:season:50039
`sport_event_id`	Sport event id expressed as: {sport_event_id}. Example: sport_event_id=sr:sport_event_id:13644241

Return to top

Push Statistics

```
curl -L "https://api.sportradar.com/soccer/trial/v4/stream/statistics/subscribe?api_key={your_api
_key}"
```

The above command returns json like this.

Provides real-time team and player match-level statistics for all live matches.

https://api.sportradar.us/soccer/`{access_level}`/`{version}`/stream/statistics/subscribe?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`your_api_key`	Your API key.

Optional Query String Parameters

Example including optional query string parameters:

```
curl -L GET 'api.sportradar.com/soccer/trial/v4/stream/statistics/subscribe?api_key={your_api_ke
y}&amp;format=json&amp;sport_event_id=sr:match:13468929'
```

In addition to the URL parameters listed above, you can filter the Statistics information with one or more of the following optional query string parameters.

Note: Optional query string parameters must be added after your API key with an ampersand (&). If you are filtering for more than one result, separate the results with a comma (,) and no spaces.

Replace placeholders with the following query parameters:

Parameter	Description
`competition_id`	Competition id expressed as: {competition_id}. Example: competition_id=sr:competition:204
`event_id`	Event type expressed as: {event_type}. Example: event_id=free_kick
`format`	Format type expressed as: {format}. Example: format=json
`season_id`	Season id expressed as: {season_id}. Example: season_id=sr:season:50039
`sport_event_id`	Sport event id expressed as: {sport_event_id}. Example: sport_event_id=sr:sport_event_id:13644241

Return to top

Probabilities Feeds

This collection of probability feeds is available for licensing as an add-on feature. They are an extension of the Season Probabilities feed in the main package which provides pre-match probabilities for the sport event winner market.

The main features of the Probabilities extension (where coverage applies) are:

- Live Probabilities that update throughout game
- Season Outright Probabilities for the Tournament Winner market
- Live Probabilities Coverage indicator for the next 24 hours

For more information or to request a Trial of this feature, please contact a Sportradar Sales representative.

The Open API specification and Syntax for using our Probabilities feeds, including examples of the payloads can be found below.

OpenAPI Specification

Live Probabilities

```
curl -X GET "https://api.sportradar.us/soccer-probabilities/trial/v4/en/schedules/live/probabilit
ies.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Provides top-level information for live matches. If probabilities are available for a match, pre-match and live probabilities will be displayed.

https://api.sportradar.com/soccer-probabilities/`{access_level}`/`{version}`/`{language_code}`/schedules/live/probabilities.`{format}`?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Live Probabilities feed, use the following URL.

https://api.sportradar.com/soccer-probabilities/trial/v4/openapi/swagger/index.html#/schedules/getScheduleLiveProbabilities

Season Outright Probabilities

```
curl -X GET "https://api.sportradar.us/soccer-probabilities/trial/v4/en/seasons/sr:season:54571/o
utright_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Provides a list of outright probabilities for each competitor from a given season.

https://api.sportradar.com/soccer-probabilities/`{access_level}`/`{version}`/`{language_code}`/seasons/`{season_id}`/outright_probabilities.`{format}`?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`season_id`	Id of a given season.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Season Outright Probabilities feed, use the following URL.

https://api.sportradar.com/soccer-probabilities/trial/v4/openapi/swagger/index.html#/seasons/getSeasonOutrightProbabilities

Sport Event Probabilities

```
curl -X GET "https://api.sportradar.us/soccer-probabilities/trial/v4/en/sport_events/sr:sport_eve
nt:17693932/sport_event_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Provides pre-match and live probabilities for a given match.

https://api.sportradar.com/soccer-probabilities/ `{access_level}` / `{version}` / `{language_code}` /sport_events/ `{sport_event_id}` /probabilities. `{format}` ?api_key= `{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Event Probabilities feed, use the following URL.

https://api.sportradar.com/soccer-probabilities/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventAdvancedProbabilities

Sport Event Upcoming Probabilities

```
curl -X GET "https://api.sportradar.us/soccer-probabilities/trial/v4/en/sport_events/upcoming_pro
babilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Provides a list of IDs for upcoming sport events in the next 24 hours.

https://api.sportradar.com/soccer-probabilities/ `{access_level}` / `{version}` / `{language_code}` /sport_events/upcoming_probabilities. `{format}` ?api_key= `{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).

`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`format`	xml or json.
`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Sport Event Upcoming Probabilities feed, use the following URL.

https://api.sportradar.com/soccer-probabilities/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventsUpcomingProbabilities

Timeline Probabilities

```
curl -X GET "https://api.sportradar.us/soccer-probabilities/trial/v4/en/sport_events/sr:sport_eve
nt:17693932/timeline_probabilities.xml?api_key={your_api_key}"
```

The above command returns xml structured like this.

Provides a timeline of pre-match and live probability changes for a given match. This also provides match information including player stats, team stats, and scoring info.

https://api.sportradar.com/soccer-probabilities/`{access_level}`/`{version}`/`{language_code}`/sport_events/`{sport_event_id}`/timeline_probabilities.`{format}`?api_key=`{your_api_key}`

Replace placeholders with the following query parameters:

Parameter	Description
`access_level`	Defines the access level of your API key as Production (production) or Trial (trial).
`version`	Version number of the API you are accessing (Current Version: v4).
`language_code`	2 letter code for supported languages Chinese - simplified (zh), Chinese - traditional (zht), Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Hindi (hi), Indonesian (id), Italian (it), Japanese (ja), Portuguese (pt), Russian (ru), Serbian (sr), Serbian Latin (srl), Spanish (es), Thai (th), Turkish (tr) Click here for a tabular list of available languages per competition.
`sport_event_id`	Id of a given sport event.
`format`	xml or json.

`your_api_key`	Your API key.

To retrieve the OpenAPI Schema Definition for the Timeline Probabilities feed, use the following URL.

https://api.sportradar.com/soccer-probabilities/trial/v4/openapi/swagger/index.html#/sport_events/getSportEventTimelineProbabilities

Frequently Asked Questions

Q: What leagues or tournaments do you cover for soccer?

A: You can find all the leagues we cover, as well as a breakdown of data offered, via our Coverage Matrix.

Q: Are simulations available for Soccer v4?

A: Simulations for Soccer v4 are not currently available.

Q: How are friendlies handled in the Summary feeds?

A: The Summary feeds will return all Friendlies that have been played the last two weeks and that will be played the next 4 weeks.

Q: What is the Date format?

A: When we present date only values we present these in the ISO 8601 standard format.
ex: 2013-04-03
We use these for attributes that have date and no time (such as birthdate). For more information:
https://en.wikipedia.org/wiki/ISO_8601

Q: What time zone are the date time fields presented in?

A: All of our Date/Time attributes are in UTC, presented in the ISO 8601 standard format.
ex: 2013-04-03T18:15:00+00:00
For more information: https://en.wikipedia.org/wiki/ISO_8601

Q: When is coverage information added to the feeds?

A: On the first day of the season.

Q: How can I find the values for various enum data points within the API?

A: Many enum values are listed in the FAQ below. For the most up-to-date values, please see the Schema section of the OpenAPI specification here:

https://api.sportradar.com/soccer/trial/v4/openapi/swagger/index.html

Q: What are the valid sport_event_status – status values?

A: Here are the valid status values and their definitions:

- not_started – The match is scheduled to be played
- started - The match has begun
- live – The match is currently in progress
- postponed – The match has been postponed to a future date
- suspended - The match has been suspended
- match_about_to_start - The match is about to begin
- delayed – The match has been temporarily delayed and will be continued. Typically appears prior to match start
- interrupted - The match began, but coverage has stopped for a short time. Note that match scores may not be updated during this period, the last recorded match score will be displayed instead
- cancelled – The match has been cancelled and will not be played
- ended – The match is over
- closed – The match results have been confirmed

Q: What are the valid sport_event_status – match_status values?

A: Here are the valid match status values and their definitions:

- not_started – The match is scheduled to be played
- started - The match has begun
- 1st_half – The match is in the first half
- 2nd_half – The match is in the second half
- overtime – The match is in overtime
- 1st_extra – The match is in the first extra period
- 2nd_extra – The match is in the second extra period
- awaiting_penalties – Waiting for announcement of penalties
- penalties – Penalties are ongoing
- awaiting_extra_time – Waiting on referee to announce extra time
- interrupted – The match has been interrupted
- abandoned – The match has been abandoned
- postponed – The match has been postponed to a future date
- start_delayed – The match has been temporarily delayed and will be continued
- cancelled – The match has been cancelled and will not be played
- halftime – The match is in halftime
- extra_time_halftime – The match is in extra time halftime
- ended – The match has ended
- aet – The match has ended after extra time
- ap – The match has ended after penalties

Q: What are the valid period type and period name values for events?

A: Here are the valid period name and period type values:

- regular_period
- overtime
- penalties
- interrupted

Q: What are the valid lineup types (player position) values?

A: Here are the valid lineup type

- goalkeeper
- defender
- midfielder
- forward

Q: What are the valid lineups descriptions (player tactical position) values?

A: Here are the valid lineups descriptions:

- goalkeeper
- right back
- central defender
- left back
- right winger
- central midfielder
- left winger
- striker

Q: What are the valid current outcome values?

A: Here are the valid current outcome types

- AFC Champions League
- AFC Cup
- CAF Confederation Cup
- Champions League
- Champions League Qualification
- Champions Round
- Championship Round
- Club Championship
- Conference League Qualification
- Copa Libertadores
- Copa Libertadores Qualification
- Copa Sudamericana
- Cup Winners
- Eliminated
- European Cup
- Final Four
- Final Round
- Finals
- Group Matches
- International Competition
- Main Round

- Next Group Phase
- Placement Matches
- Playoffs
- Preliminary Round
- Promotion
- Promotion Playoff
- Promotion Playoffs
- Promotion Round
- Qualification Playoffs
- Qualified
- Qualifying Round
- Relegation
- Relegation Playoff
- Relegation Playoffs
- Relegation Round
- Semifinal
- Top Six
- UEFA Conference League
- UEFA Conference League Qualification
- UEFA Cup
- UEFA Cup Qualification
- UEFA Europa League
- UEFA Europa League Qualification
- UEFA Intertoto Cup

Q: How do I find out the coverage for a particular match?

A: Within the Summaries, Lineups, and Timeline feeds, find the node for “coverage”. Coverage nodes have three types: season level, group level, and sport_event level.

- The season level describes data coverage you can expect for matches involved in that given season.
- The group level is similar because there exists competitions where coverage levels differ at different stages or in different groups - mostly cup competitions.
- The sport_event level, this exists to describe the data depth of a specific match within the group and season.

Note: There are occasions when the sport_event coverage can vary from the anticipated season coverage and this node exists to highlight that in that event to assist in handling any discrepancies.

Q: How is coverage for a particular match defined?

A: The coverage nodes contain classifications of data types which are expressed as Boolean values or denoted as “live” or “post”, for example: ballspotting="true" commentary="true" extended_play_by_play="true" extended_player_stats="true" extended_team_stats="true" fun_facts="true" goal_scorers="true" scores="live" basic_play_by_play="true" basic_player_stats="true" basic_team_stats="true" lineups="true" deeper_player_stats="true" deeper_team_stats="true".

Q: What are the possible pitch values?

A: Here are the possible pitch values:

- good
- medium
- bad

Q: What are the possible weather conditions?

A: Here are the possible weather conditions:

- indoor
- good
- medium
- bad
- extreme

Q: What are the possible event types?

A: Here are all of the possible event types we log:

- break_start
- canceled_decision_to_var
- corner_kick
- decision_to_var
- decision_to_var_over
- free_kick
- goal_kick
- injury
- injury_return
- injury_time_shown
- match_ended
- match_started
- offside
- penalty_awarded
- penalty_kick
- penalty_missed
- penalty_shootout
- period_score
- period_start
- player_back_from_injury
- possible_decision_to_var
- red_card
- save
- score_change
- shot_off_target
- shot_on_target
- shot_saved
- substitution

- throw_in
- video_assistant_referee
- video_assistant_referee_over
- yellow_card
- yellow_red_card

* Availability of Video Assistant Referee is subject to the VAR capabilities of the league.

Q: How does the possible_goal event type work?

A: This event will occur immediately if one team scores a goal or if they are in a very clear scoring opportunity (1on1 with the goalkeeper, clear shot at an empty net, etc).

Q: What are the possible values for video assistant referee?

A: Here are the valid values:

- goal
- penalty
- red_card
- no_red_card
- no_penalty
- no_goal

Q: What are the possible values for video assistant referee over?

A: Here are the valid values:

- call_stands
- call_overturned

Q: What is the scale of the X Y coordinates?

A: The pitch we use is 100 by 100. Here is a layout of the pitch:

0,0		***X***		**100,0**
Y		50,50		
0,100				100,100

X = Horizontal position on the pitch. X is a number between 0 and 100. The reference point 0 is at the home team's goal.

Y = Vertical position on the pitch. Y is a number between 0 and 100. The reference point 0 is on the top of the pitch where the home team's goal is on the left hand side.

Q: What are the possible values for referee_assistant type?

A: Here are the valid referee assistant types:

- first_assistant_referee
- second_assistant_referee
- fourth_official
- video_assistant_referee
- first_additional_assistant
- second_additional_assistant
- third_additional_assistant

Q: How does the ball location attribute work?

A: Our scouts mark down the x (lateral) and y (longitudinal) coordinates as observed on the pitch. The data can come in sporadically as events on the field play out, but new `ball_location` data is potentially available every 1 sec. This is only available for matches with `ballspotting="true"`. The element `ball_locations` stores the last four known ball locations, after which the data is not available unless it corresponds with another event in the timeline such as `throw_in`, `shot_on_goal` etc. The `ball_location` order illustrates the most recent location as 4 and the oldest location as 1.

Q: What are markets and what are the different markets?

A: Markets is something you can bet on that we provide probabilities for. Over time we intend to provide more and more markets in the API. Currently the only market we provide is 3-way (will the home team win? Or the away team? Or will it be a draw?).

Q: What are the possible outcomes?

A: Different markets have different outcomes. The available markets are currently:

- home_team_winner
- away_team_winner
- draw

Q: Why don't I see commentary in the timeline when the coverage has the "commentary" sport_event_property as "true"?

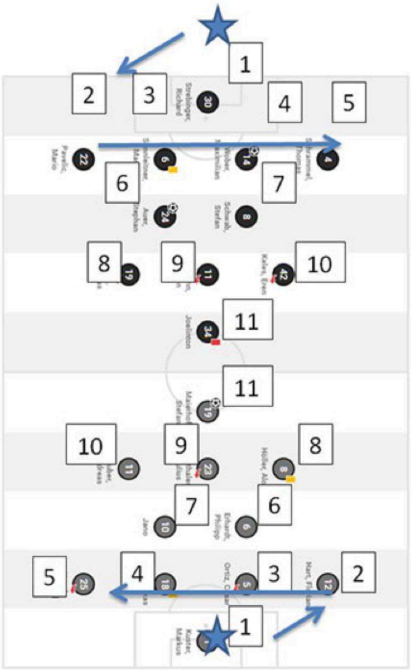
A: Commentary is only available in the feed for a fixed amount of time (14 days typically). The attribute, which denotes that commentary is or was available, remains even after the commentary is removed.

Q: When are fun facts added to the Sport Event Fun Facts feed?

A: Fun facts appear in the feed starting 7 days before a match and are available for a fixed amount of time (14 days typically).

Q: How is the order value in the Lineups feed organized?

A: Order number 1 is always the goalie (star marking) and formations as well as numbering should start with the goalkeeper. In the example diagram the formation 4-2-3-1 is used.



4 is the number of players in the line in front of the goalkeeper, then comes the line with 2 players and so on. Numbering in every line starts at the right-hand side of the goalkeeper – this causes the numbering to be mirrored for home and away team.

Q: What are the possible reasons for a player to appear in the Missing Players feed?

A: Listed below are the reasons why a player may appear in the missing players feed:

- injured
- ill
- other
- suspension

Q: What are the possible values for stage – phase in the feeds?

A: Listed below are the values for phase:

- 1st_part_of_season_1st_leg
- 2nd_part_of_season_2nd_leg
- 3rd_round
- champions_round
- conference
- division
- final_eight
- final_four
- final_phase
- final_round
- final_stage

- grand_final
- grand_finals
- group_phase_1
- group_phase_2
- knockout_stage
- main_round_1
- main_round_2
- none
- placement_matches
- placement_matches_13_to_16
- placement_matches_5_to_8
- placement_matches_9_to_12
- placement_matches_9_to_16
- playoffs
- playout
- pre-season
- preliminary_round
- president_cup
- promotion_playoffs
- promotion_round
- qualification
- qualification_playoffs
- qualification_to_allsvenskan
- regular season
- relegation_playoffs
- relegation_promotion
- relegation_promotion_round
- relegation_round
- stage_1
- stage_1 no_stats
- stage_2
- stage_2 no_stats
- stage_3
- uefa_europa_league_playoffs

Q: How long is full match data available for matches in the API?

A: Match data is archived after one year and you will only be able to service basic score information from the API feeds. A historical statistics API for Soccer is on the roadmap, but no ETA is available at this time.

Q: What are the possible values for event – time_description in the feeds?

A: Listed below are some of the values for time descriptions on yellow/red cards.

- pre_match
- half_time

- post_match

Q: What are the possible values for list – type in the leaders feeds?

A: Listed below are the values for list - type. Note: Not all values may be available for each season, based on the coverage available for that season.

- points
- goals
- assists
- red_cards
- yellow_cards
- yellow_red_cards
- own_goals
- shots_on_target
- shots_off_target
- goals_by_head
- goals_by_penalty
- clearances
- interceptions
- chances_created
- crosses_successful
- passes_successful
- long_passes_successful
- tackles_successful
- clean_sheets
- penalties_saved
- dribbles_completed
- loss_of_possession
- minutes_played

Q: What are the possible values for event – card_description in the timeline feeds?

A: Listed below are the values for event – card_description.

- pre_match
- half_time
- post_match
- player_on_bench
- first_half
- second_half
- during_penalty_shootout

Q: Why does the coverage of past seasons not match the data in the feeds?

A: Coverage properties are set at a competition level and only reflect the current or last season of that competition. Previous seasons may have greater or lesser coverage.

Q: When does the Lineups Availability update in the feeds?

A: The lineups_availability data is updated 30 days before the sport event is scheduled to begin.

Q: What are the possible values for sport_event_properties – lineups_availability in the summary and timeline feeds?

A: Listed below are the values for sport_event_properties – lineups_availability:

- pre
- post

Q: How do you define Team Statistics and Player Statistics as sport_event_properties in terms of coverage?

A: We define those properties as:

Team Statistics:

basic_team_stats: yellow_cards, yellow_red_cards, red_cards.

deeper_team_stats: corner_kicks, shots_total, shots_on_target, shots_off_target, shots_blocked, ball_possession, free_kicks, offsides, goal_kicks, throw_ins, shots_saved, fouls, and injuries.

Player Statistics:

basic_player_stats: goals_scored, yellow_cards, yellow_red_cards, red_cards, own_goals, assists, substituted_in, and substituted_out.

deeper_player_stats: offsides, corner_kicks, shots_on_target, shots_off_target, and shots_blocked.

Q: How do you define basic and deeper play by play in terms of coverage?

A: Basic_play_by_play includes score_change, cards, and substitutions. Deeper_play_by_play includes all other event types.

Q: How are replay cup matches handled in the feeds?

A: Within the Summary, Timeline, or Lineups feeds you can locate the round data for a given match. In that round data you can find the number of matches in the cup round (cup_round_number_of_sport_events) and the number of the given match in the cup round (cup_round_sport_event_number). The values for cup_round_sport_event_number are detailed below:

- 1 = Replay
- 2 = 1st Replay
- 3 = 2nd Replay

Q: How are group IDs delivered in the stage array with the various types?

A: With the type of "league" they will have a sr:league prefix. With the type of "cup" they will have a sr:cup prefix.

Q: Which regions are covered with TV Channel data?

A: We offer Network TV data for the United States. This will be available for MLS, World Cup, EPL, UEFA Champions League, Bundesliga, Liga MX, and Gold Cup.

Q: Why does the coverage of a cup competition not match the data in the feeds?

A: For cup competitions, coverage levels may vary from the early rounds to latter stages. Coverage properties are set at a competition level and will display the best coverage we offer for a sport event in this competition.

Q: How are the seasonal competitor statistics handled within the feed?

A: Statistics from any qualification rounds will not be displayed. The feed will only display data from the main competition.

Q: How do you define the key data points?

Data Points	Type of Data Point	Definition
Goal	regular	A goal is assigned to a player of the attacking team if his shot taken by foot, head or any other part of the body leads to a score. All attempts are counted. Uncontrolled deflections resulting in a goal will be attributed to the player who kicked the ball, not the player who deflects the ball.
	own goal	When a defender kicks or heads the ball into the defender's goal with the intent of making a pass to a teammate or of clearing the ball and it is obvious that no other player of the attacking team tried a shot on target, a goal is recorded as an "own goal". An "own goal" counts towards the offensive team's season statistics. No assist shall be given to the attacking team. If an attacking player shots on goal and the defender deflects the ball into the net, in a controlled action it will be deemed an own goal. If two players of the defending team touched the ball, the player who last touched it is the own goal scorer.
Assist	in general	An assist considered to be a contribution by a player which directly helps a teammate to score a goal. In order to be considered for an assist a player needs to have given the last successful pass. However, the last successful pass does not necessarily automatically constitute an assist.
	counted	• Includes corners and throw ins • A cross by a team member is converted into a goal • Passes/crosses, which are slightly deflected and then lead to a team member score a goal
	not counted	• Where goals resulting from penalties are concerned, the player who is fouled in the area receives no assist point • shot hits the post/bar and then leads to a goal by a team member • The goal scorer made considerable self-effort in order to score
Throw-In		If the ball crosses any of the side lines by its diameter, the game will continue with a throw-in.

Free kicks	team statistics	In accordance with the rules, a free kick stat counted for the team who executes a free kick. Direct or indirect free kicks are all counted as free kicks.
Goal kick	team statistics	Total number of kicks awarded to the team as a result of the ball traveling out of bounds over the goal line of the defending team
Throw ins	team statistics	Total number of thrown in events for a team
Offsides	team statistics	Total number of offside infringements awarded against a team.
Fouls	team statistics	Total number of fouls awarded against a team (including those which draw cards)
Corner Kicks	team statistics	Total number of corner kicks taken by a team.
Shots Saved	team statistics	Total number of goal keeper saves attributed to a team.
Ball Possession	team statistics	Ball possession is the amount of time a team has possession of the ball. It is calculated in percentage, and the total amount of time the match is being played is considered 100%.
Shots	On Target	This is an intentional shot on target by a player, trying to score a goal by foot, head or any other part of the body. The result of a shot on goal is either a goalkeeper save, a clearance by any player of the defending team, a blocked shot by a player of the attacking team (which happens very rarely) or a goal. A shot on goal is slightly deflected by a player of the attacking team into the net, a goal will be awarded to the player who initially shot the ball. The corresponding team stat for this is “shots_on_target”.
	Off Target	A shot which was aimed towards the goal but missed. These shots do not necessarily have to cross the goal line. A ball which hits the post or the bar without touching anyone and is still in play, is considered a shot off goal. The corresponding team statistic for this data point is “shots_off_target”.
	Blocked	A shot which was directed on target but was blocked by a player of the opposing team.

Q: What prompts a match to appear in Sport Events Updated?

A: Changes to score, match status, or schedule in last 24 hours would cause a match to display in this endpoint.

Q: How will a sport event behave when it is not covered with live scores?

A: When a sport_event is not covered live, the status and match_status will remain as not_started until results are entered post-match.

Q: How is the minutes_played statistic calculated?

A: Minutes played is calculated based on 90 minutes in the match. We do not include stoppage_time.

Q: Do you cover VAR events?

A: VAR events are supported with events 'video_assistant_referee' & 'video_assistant_referee_over' however we cannot guarantee the accuracy or frequency of these events.

Q: How are "live" feeds handled in the API?

A: Sport Events appear in the feed 10 minutes before the scheduled start time and are removed 10 minutes after the Sport Event is ended.

Q: What is the probabilities package?

A: The probabilities package is an add-on set of feeds that are an extension of the Season Probabilities feed in the main package (which already provides pre-match probabilities for the sport event winner market). The main features of the Probabilities extension are: Live Probabilities that update throughout game, Season Outright Probabilities for the Tournament Winner market, and Live Probabilities Coverage indicator for the next 24 hours.

Q: Is there Live Probability coverage for every game you cover in the Soccer API?

A: Live Probability coverage depends on a number of factors and we can't guarantee that every game will be covered for any given league, however, Sportradar covers tens of thousands of sport events across hundreds of leagues. For individual competition coverage, contact sales@sportradar.com

Q: What are the possible standings types I can observe in the Standings feed?

A: Listed below are the values for standing – type.

- total
- home
- away
- first_half_total
- first_half_home
- first_half_away
- second_half_total
- second_half_home
- second_half_away

Q: Why can't I find a particular match in the Daily Summaries, Live Summaries, Season Summaries or Sport Events Updated feeds?

A: These endpoints support pagination and return 200 entries by default. To return more matches, an additional query string parameter must be used after your API key. For example, appending *&start=200* will return the next 200 entries per page, *&start=400* will return the next 200, and so on.

Q: Do you have player transfer data available?

A: We use roles from player profiles to create a Season Transfers endpoint. This will display any player recently assigned to a team in a season that is covered by our squad coverage and has `player_transfer_history="true"`.

This can include youth players that have recently been added to a matchday squad. If there is no previous club within 10 days, then this will be understood to be a free agent and therefore no `from_competitor` will display. For `transfer_date`, we use multiple sources and cannot guarantee the accuracy.

Q: Why does the change attribute not update for Live Standings?

A: The ‘change’ attribute is only available post-match and will display movement from the previous gameweek/round.

Q: When will lineups confirmed show as true?

A: Lineups information will display as they are entered. Once completed, they are then confirmed at which point the attribute for lineups confirmed will update to `true`. This will only show for matches that have `lineups_availability="pre"`

Q: Are Live Standings available?

A: Live standings are delivered by default in the Standings endpoint. Live standings use an automatic set of tiebreaker rules that are calculated based on the scores while matches are in progress. We recommend adding the parameter `live=false` to a Standings request to retrieve standings when matches are not in progress. This will consider any additional tie-break criteria that a competition may use if competitors are level.

Q: How do I locate the TTL (Time to Live)/cache on an API endpoint?

A: The cache (in seconds) can be accessed in the returned header information on each RESTful API call, under `cache-control`. ex. `cache-control: max-age=1, public, s-maxage=1` or `cache-control: public, must-revalidate, max-age=120`

Q: What are the possible values for `cup_round` – `state` in the Season Links feed?

A: Listed below are the values and definitions for `cup_round` - `state`. These can be leveraged to determine the status of a cup round.

- `empty` - A matchup has been created but neither the match details nor the competitors are known.
- `unseeded_fixture` - Match details are known but competitors are unknown.
- `partial_seeded` - One competitor is known.
- `partial_seeded_fixture` - Match details and one competitor are known.
- `seeded` - Both competitors are known.
- `seeded_fixture` - Match details and both competitors are known.
- `unstarted` - Match(es) have been added.
- `on_going` - The first match has started.
- `decided` - The last match has ended.
- `winner` - The winner is known.
- `cancelled` – The matchup has been cancelled.

Q: What are the possible values for match situation status (`match_situation.status`)?

A: Listed below are the values and definitions for `match_situation` - `status`. These can be leveraged to determine the status of a ball in play.

- `safe` - Team in possession of the ball is inside their defensive half
- `dangerous` - Team in possession is in the opponent’s half but not near the penalty box
- `attack` - Team in possession is in the opponent’s half, near the penalty box

Q: Why is goals scored the only statistic available in Season Leaders and Season Competitor Statistics for some competitions?

A: This is determined by the assists="false" property in the Season Info feed which indicates that stats below goals_scored are not guaranteed for the competition in question.

Return to top

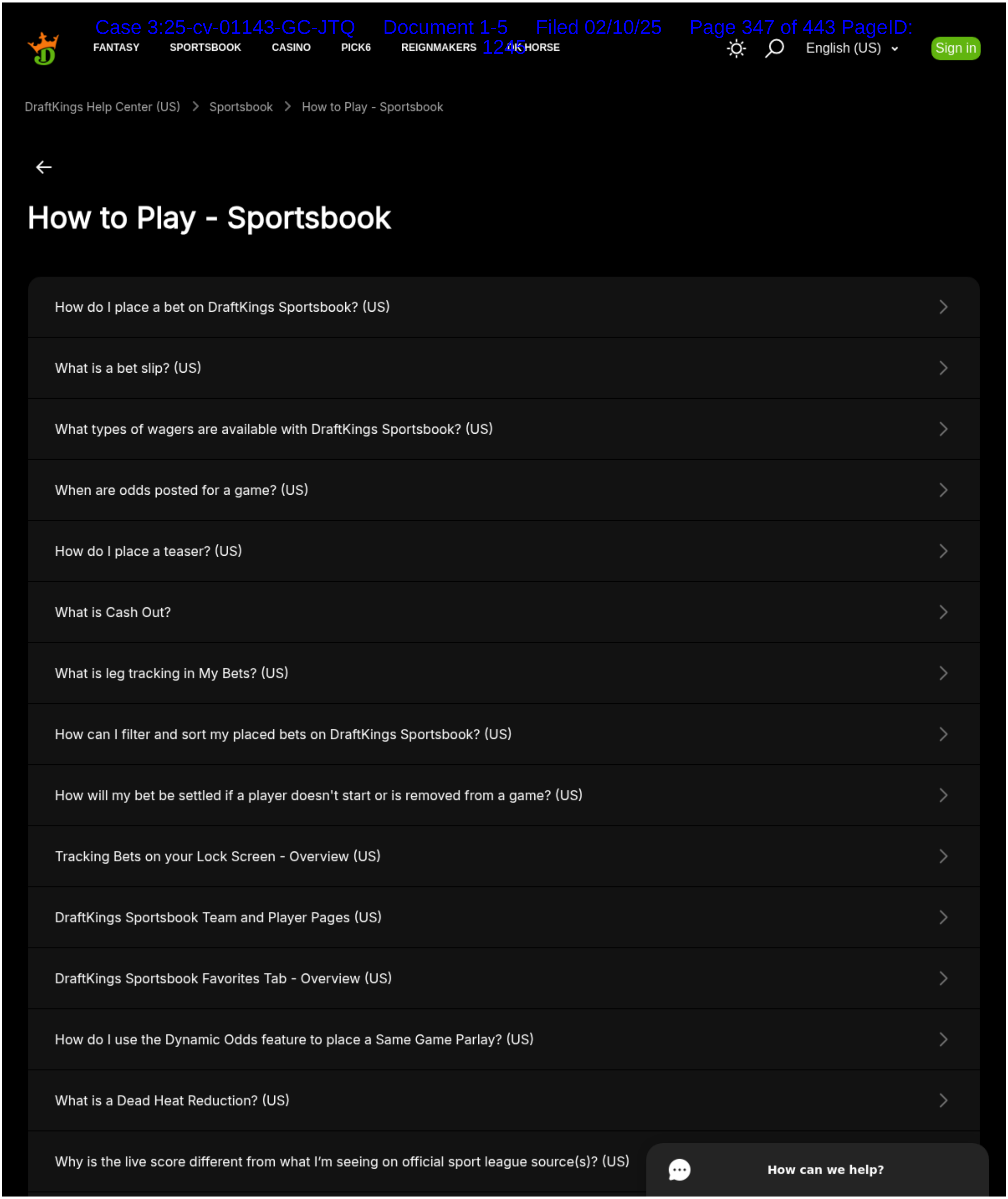
SPORTRADAR
150 South 5th St. Suite 400
Minneapolis, MN 55402
USA

us-sales@sportradar.com
us-jobs@sportradar.com
Main: 612-361-4100
Sales: 612-361-4102
Terms and Conditions

SPORTRADAR IS SOCIAL

Facebook Technology Partner

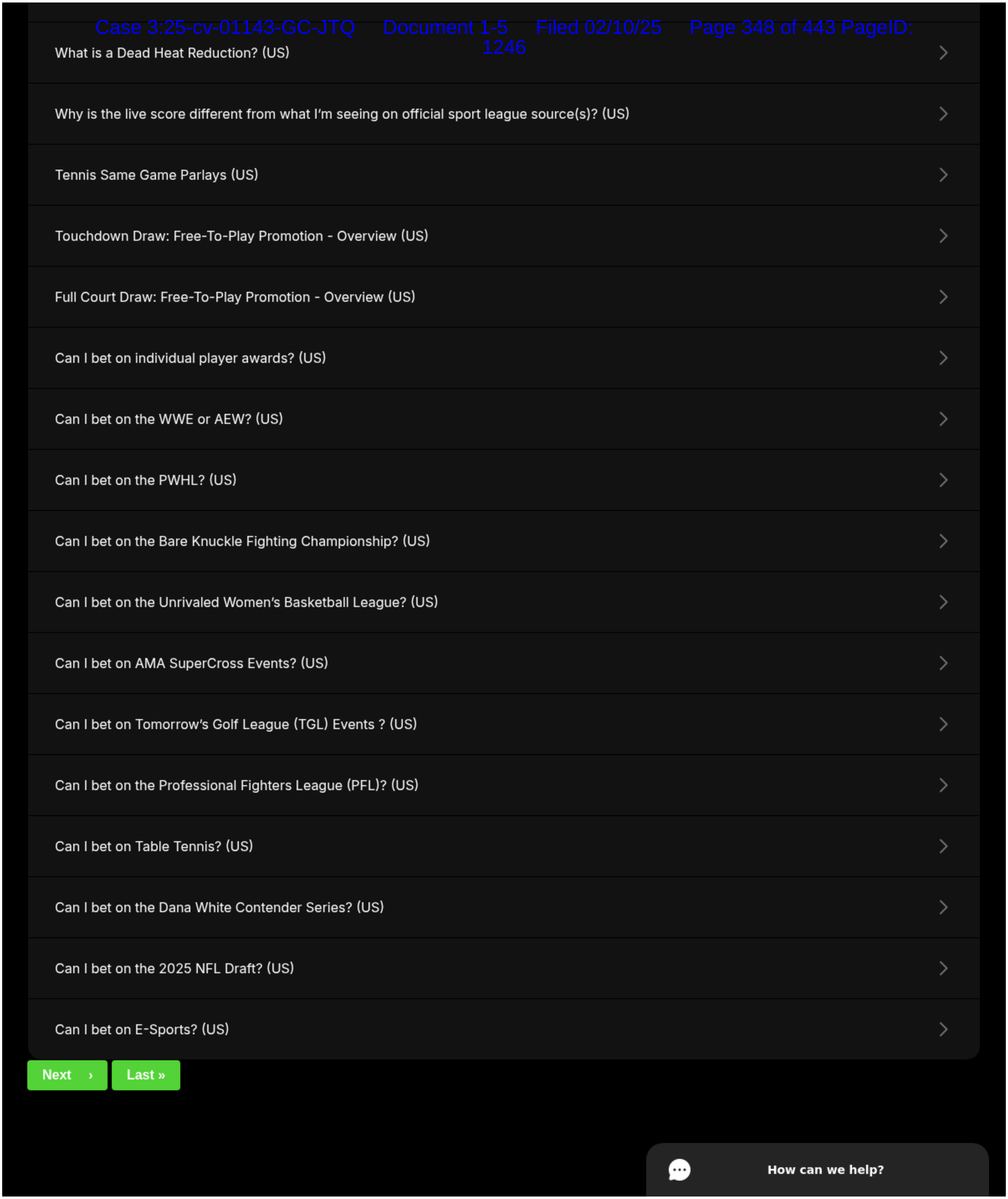
Exhibit 58



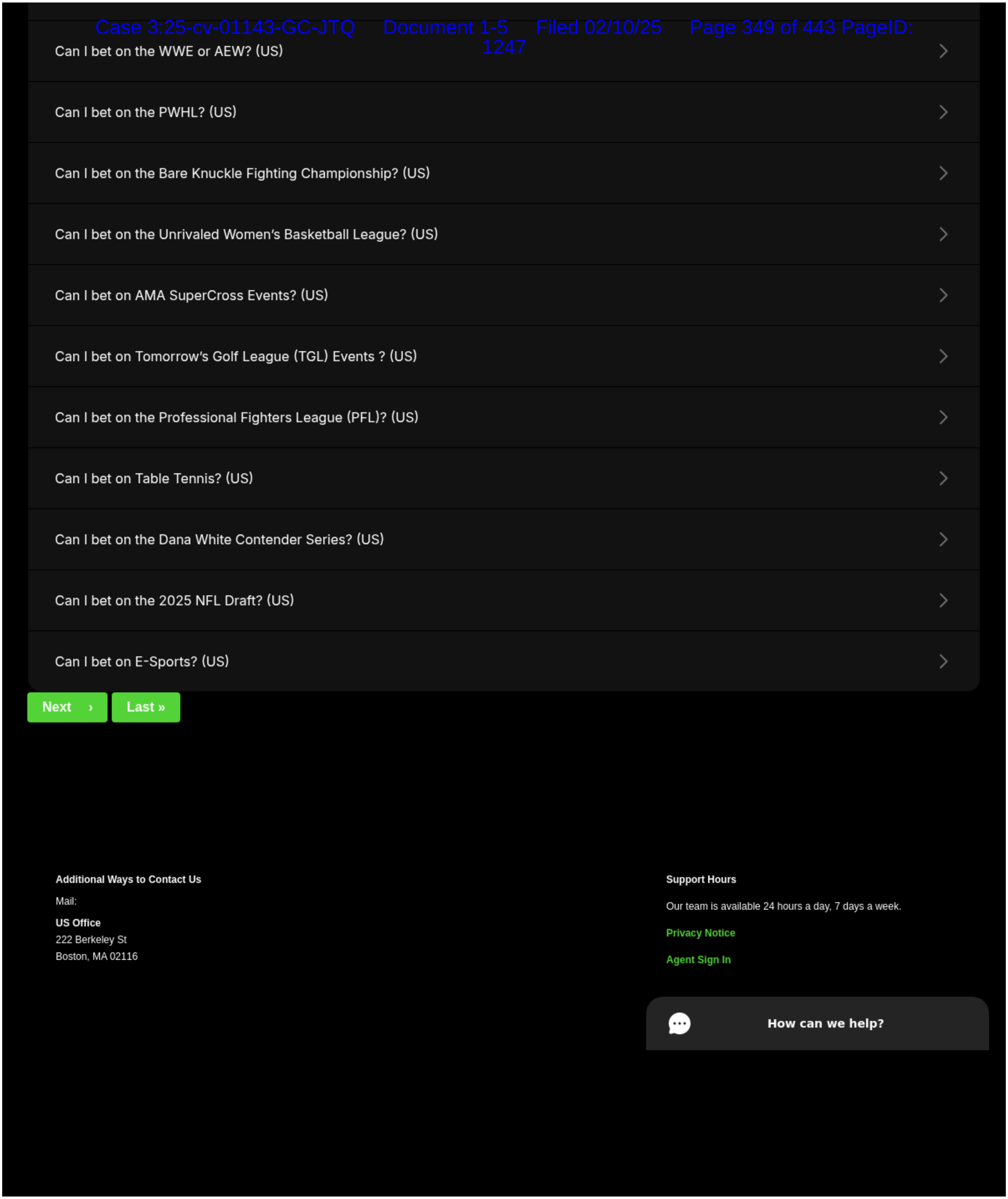
DraftKings Help Center (US) > Sportsbook > How to Play - Sportsbook

How to Play - Sportsbook

- How do I place a bet on DraftKings Sportsbook? (US)
- What is a bet slip? (US)
- What types of wagers are available with DraftKings Sportsbook? (US)
- When are odds posted for a game? (US)
- How do I place a teaser? (US)
- What is Cash Out?
- What is leg tracking in My Bets? (US)
- How can I filter and sort my placed bets on DraftKings Sportsbook? (US)
- How will my bet be settled if a player doesn't start or is removed from a game? (US)
- Tracking Bets on your Lock Screen - Overview (US)
- DraftKings Sportsbook Team and Player Pages (US)
- DraftKings Sportsbook Favorites Tab - Overview (US)
- How do I use the Dynamic Odds feature to place a Same Game Parlay? (US)
- What is a Dead Heat Reduction? (US)
- Why is the live score different from what I'm seeing on official sport league source(s)? (US)



- What is a Dead Heat Reduction? (US)
- Why is the live score different from what I'm seeing on official sport league source(s)? (US)
- Tennis Same Game Parlays (US)
- Touchdown Draw: Free-To-Play Promotion - Overview (US)
- Full Court Draw: Free-To-Play Promotion - Overview (US)
- Can I bet on individual player awards? (US)
- Can I bet on the WWE or AEW? (US)
- Can I bet on the PWHL? (US)
- Can I bet on the Bare Knuckle Fighting Championship? (US)
- Can I bet on the Unrivaled Women's Basketball League? (US)
- Can I bet on AMA SuperCross Events? (US)
- Can I bet on Tomorrow's Golf League (TGL) Events ? (US)
- Can I bet on the Professional Fighters League (PFL)? (US)
- Can I bet on Table Tennis? (US)
- Can I bet on the Dana White Contender Series? (US)
- Can I bet on the 2025 NFL Draft? (US)
- Can I bet on E-Sports? (US)



Can I bet on the WWE or AEW? (US)

Can I bet on the PWHL? (US)

Can I bet on the Bare Knuckle Fighting Championship? (US)

Can I bet on the Unrivaled Women's Basketball League? (US)

Can I bet on AMA SuperCross Events? (US)

Can I bet on Tomorrow's Golf League (TGL) Events ? (US)

Can I bet on the Professional Fighters League (PFL)? (US)

Can I bet on Table Tennis? (US)

Can I bet on the Dana White Contender Series? (US)

Can I bet on the 2025 NFL Draft? (US)

Can I bet on E-Sports? (US)

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

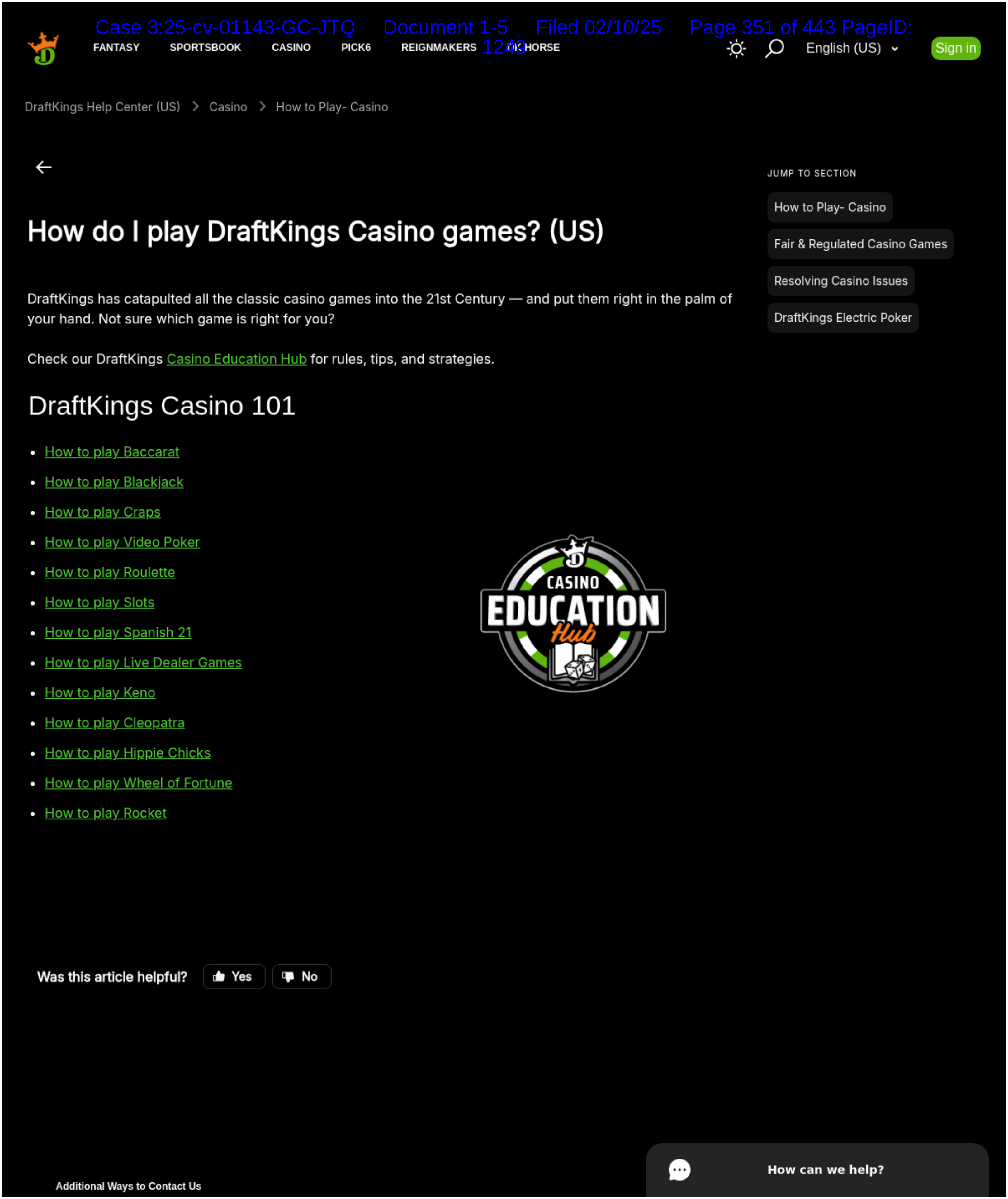
Support Hours

Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

Exhibit 59



DraftKings Help Center (US) > Casino > How to Play- Casino

How do I play DraftKings Casino games? (US)

DraftKings has catapulted all the classic casino games into the 21st Century — and put them right in the palm of your hand. Not sure which game is right for you?

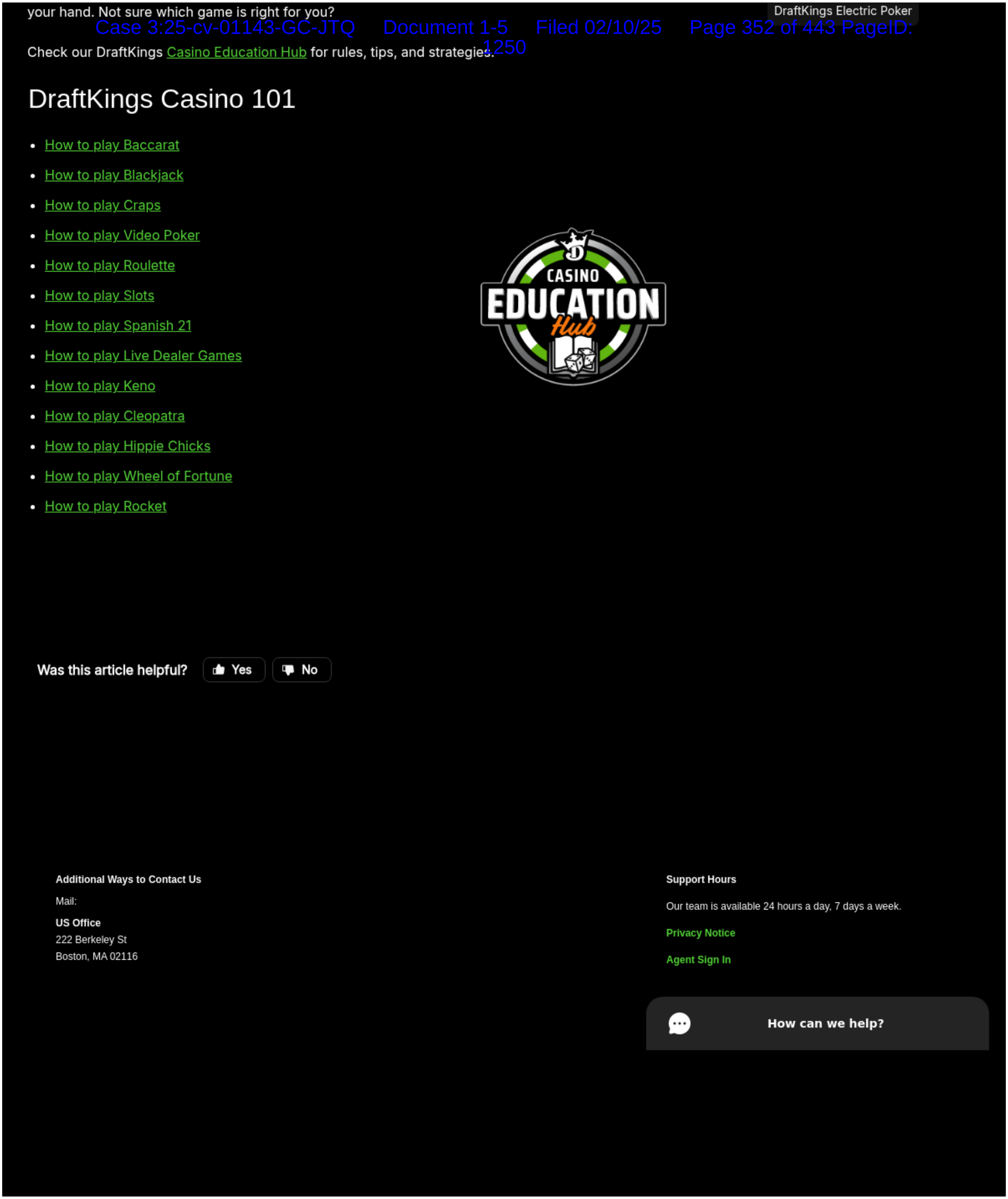
Check our DraftKings Casino Education Hub for rules, tips, and strategies.

DraftKings Casino 101

- How to play Baccarat
- How to play Blackjack
- How to play Craps
- How to play Video Poker
- How to play Roulette
- How to play Slots
- How to play Spanish 21
- How to play Live Dealer Games
- How to play Keno
- How to play Cleopatra
- How to play Hippie Chicks
- How to play Wheel of Fortune
- How to play Rocket

Was this article helpful? Yes No

Document title: How do I play DraftKings Casino games? (US) – DraftKings Help Center (US)
Capture URL: https://help.draftkings.com/hc/en-us/articles/4405146626963-How-do-I-play-DraftKings-Casino-games-US
Capture timestamp (UTC): Tue, 28 Jan 2025 20:56:57 GMT



your hand. Not sure which game is right for you?

Check our DraftKings Casino Education Hub for rules, tips, and strategies.

DraftKings Casino 101

- How to play Baccarat
- How to play Blackjack
- How to play Craps
- How to play Video Poker
- How to play Roulette
- How to play Slots
- How to play Spanish 21
- How to play Live Dealer Games
- How to play Keno
- How to play Cleopatra
- How to play Hippie Chicks
- How to play Wheel of Fortune
- How to play Rocket

Was this article helpful? Yes No

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

Document title: How do I play DraftKings Casino games? (US) – DraftKings Help Center (US)
Capture URL: https://help.draftkings.com/hc/en-us/articles/4405146626963-How-do-I-play-DraftKings-Casino-games-US
Capture timestamp (UTC): Tue, 28 Jan 2025 20:56:57 GMT

Exhibit 60

+ GET MORE FOOTBALL WITH NFL+ PREMIUM NFL+ NFL REDZONE NFL NETWORK Subscribe Now Learn More

KC 15-2 | SUPER BOWL | SUN 11:30PM | FOX | 14-3 PHI

WATCH GAMES NEWS TEAMS STATS SUPER BOWL LIX NFL Pro ↗ Tickets ↗ On Location ↗ Shop ↗ Sign In

YOUR GAME GUIDE

WAYS TO WATCH

Get a personalized view of the NFL schedule based on your location and services so you never miss a game this season.

Where do you live?

20187 (Warrenton, VA) Submit

How do you watch?

Select all services that apply.

TV / CABLE

TV Provider

CBS ESPN FOX NBC abc NFLN

STREAMING SERVICES

NFL+ | SUNDAY TICKET | peacock | Paramount+ | ESPN+ | prime video | NETFLIX

BY WEEK | Team Schedule

18 JAN. 8	WILD CARD WEEKEND JAN. 8 – JAN. 15	DIVISIONAL PLAYOFFS JAN. 15 – JAN. 22	CONFERENCE CHAMPIONSHIPS JAN. 22 – JAN. 28	SUPER BOWL JAN. 28 – JUL. 30

NATIONAL AND LOCAL GAMES

These games are available nationally or within your local market. There are a variety of ways to access these games. Games are subject to change.

MATCHUP	DATE	TIME	NETWORK
CHIEFS vs EAGLES	SUN FEB. 9	11:30 PM UTC	FOX

CONNECT TV & STREAMING ACCOUNTS

Log in to your TV and streaming providers for more detailed game access and to stream games.

Connect Accounts

TV / CABLE

TV providers include over the air broadcasts, YouTube TV, DirectTV, XFinity, Spectrum, Hulu, Dish, Sling, Fios, Fubo, Optimum etc. Selecting "TV Provider" assumes access to Fox, CBS, NBC, ABC, ESPN and NFL Network. Connect your account for game availability based on your specific tv provider subscription.

FLEXIBLE SCHEDULING

During the NFL's Flex Scheduling Windows, games initially scheduled for Sunday Night Football (on NBC), Monday Night Football (ESPN or ABC), and Thursday Night Football (on Amazon Prime) are tentatively scheduled and subject to change.

TBD GAMES

In Week 17, three of five designated matchups will be played on Saturday. Specific dates and start times for the designated Week 17 matchups will be determined and announced at a later date during the season. For Week 18, the final weekend of the season, the scheduling of the Saturday, Sunday afternoon, and the Sunday night games is not assigned. The schedule for Week 18 will be announced at the conclusion of Week 17.

AFC

NFC

Connect Accounts

TV / CABLE

TV providers include over the air broadcasts, YouTube TV, DirectTV, XFinity, Spectrum, Hulu, Dish, Sling, Fios, Fubo, Optimum etc. Selecting "TV Provider" assumes access to Fox, CBS, NBC, ABC, ESPN and NFL Network. Connect your account for game availability based on your specific tv provider subscription.

FLEXIBLE SCHEDULING

During the NFL's Flex Scheduling Windows, games initially scheduled for Sunday Night Football (on NBC), Monday Night Football (ESPN or ABC), and Thursday Night Football (on Amazon Prime) are tentatively scheduled and subject to change.

TBD GAMES

In Week 17, three of five designated matchups will be played on Saturday. Specific dates and start times for the designated Week 17 matchups will be determined and announced at a later date during the season. For Week 18, the final weekend of the season, the scheduling of the Saturday, Sunday afternoon, and the Sunday night games is not assigned. The schedule for Week 18 will be announced at the conclusion of Week 17.

AFC

NFC

General & Legal

- Support
- Privacy Policy
- Terms & Conditions
- Subscription Terms & Conditions
- Accessibility
- Ad Choices
- Your Privacy Choices
- Cookie Settings
- Sitemap

NFL Culture

- Careers
- In the Community
- Inspire Change
- NFL HBCU
- Por La Cultura
- Play Football
- Play 60
- NFL Origins

NFL Ecosystem

- NFL Football Operations
- NFL Shop
- NFL Films
- On Location
- Pro Football Hall of Fame
- USA Football
- NFL Extra Points Credit Card
- NFL Ticket Exchange
- NFL Ticket Network
- NFL Auction
- Flag Football
- Activate - CTV

Media

- NFL Communications
- Media Guides
- Record & Fact Book
- Rule Book
- Licensing

Players

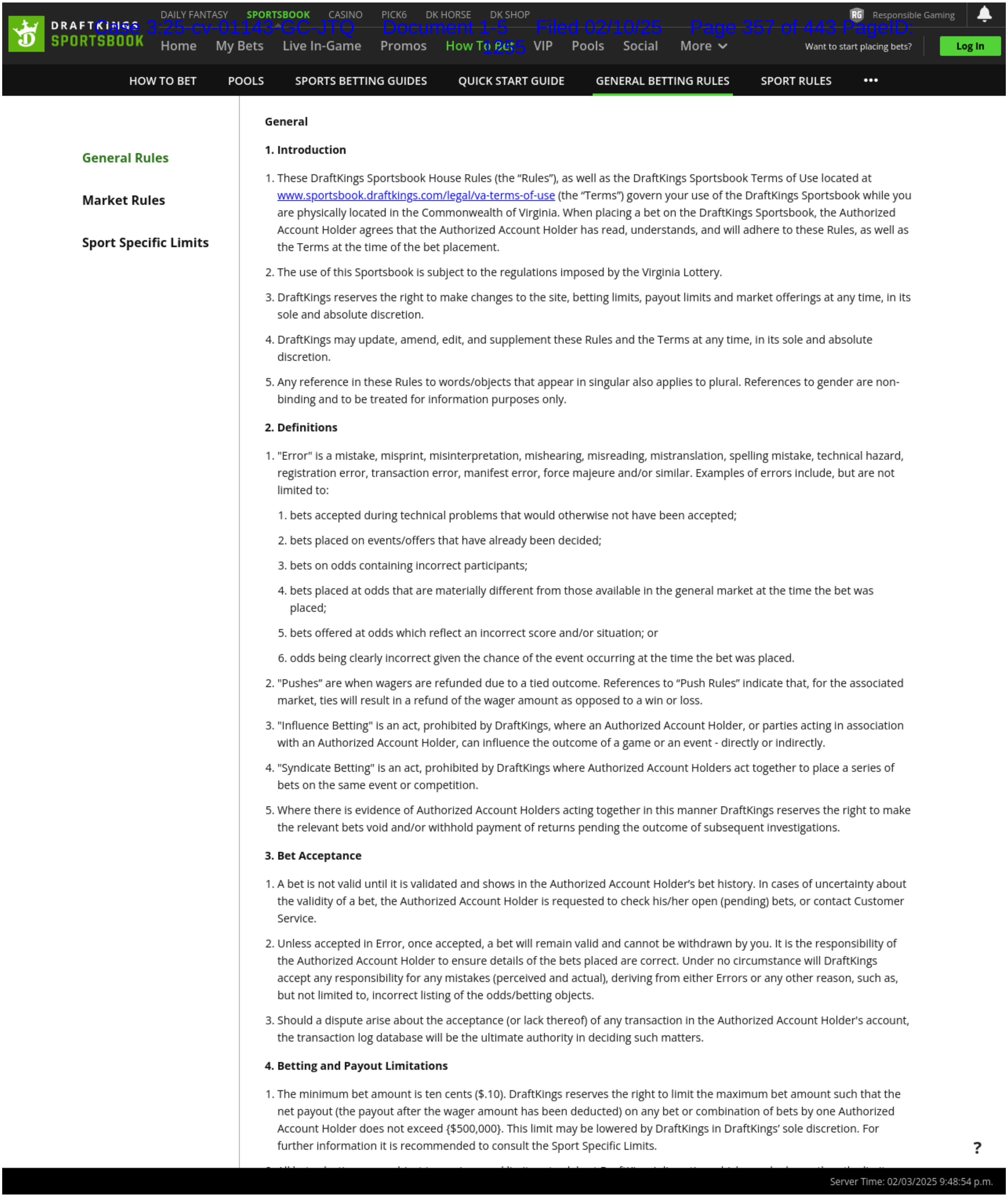
- NFL Health & Safety
- Player Engagement
- NFL Alumni Association
- NFL Player Care

Download the App

Download on the App Store

GET IT ON Google Play

Exhibit 61



General

1. Introduction

1. These DraftKings Sportsbook House Rules (the “Rules”), as well as the DraftKings Sportsbook Terms of Use located at www.sportsbook.draftkings.com/legal/va-terms-of-use (the “Terms”) govern your use of the DraftKings Sportsbook while you are physically located in the Commonwealth of Virginia. When placing a bet on the DraftKings Sportsbook, the Authorized Account Holder agrees that the Authorized Account Holder has read, understands, and will adhere to these Rules, as well as the Terms at the time of the bet placement.
2. The use of this Sportsbook is subject to the regulations imposed by the Virginia Lottery.
3. DraftKings reserves the right to make changes to the site, betting limits, payout limits and market offerings at any time, in its sole and absolute discretion.
4. DraftKings may update, amend, edit, and supplement these Rules and the Terms at any time, in its sole and absolute discretion.
5. Any reference in these Rules to words/objects that appear in singular also applies to plural. References to gender are non-binding and to be treated for information purposes only.

2. Definitions

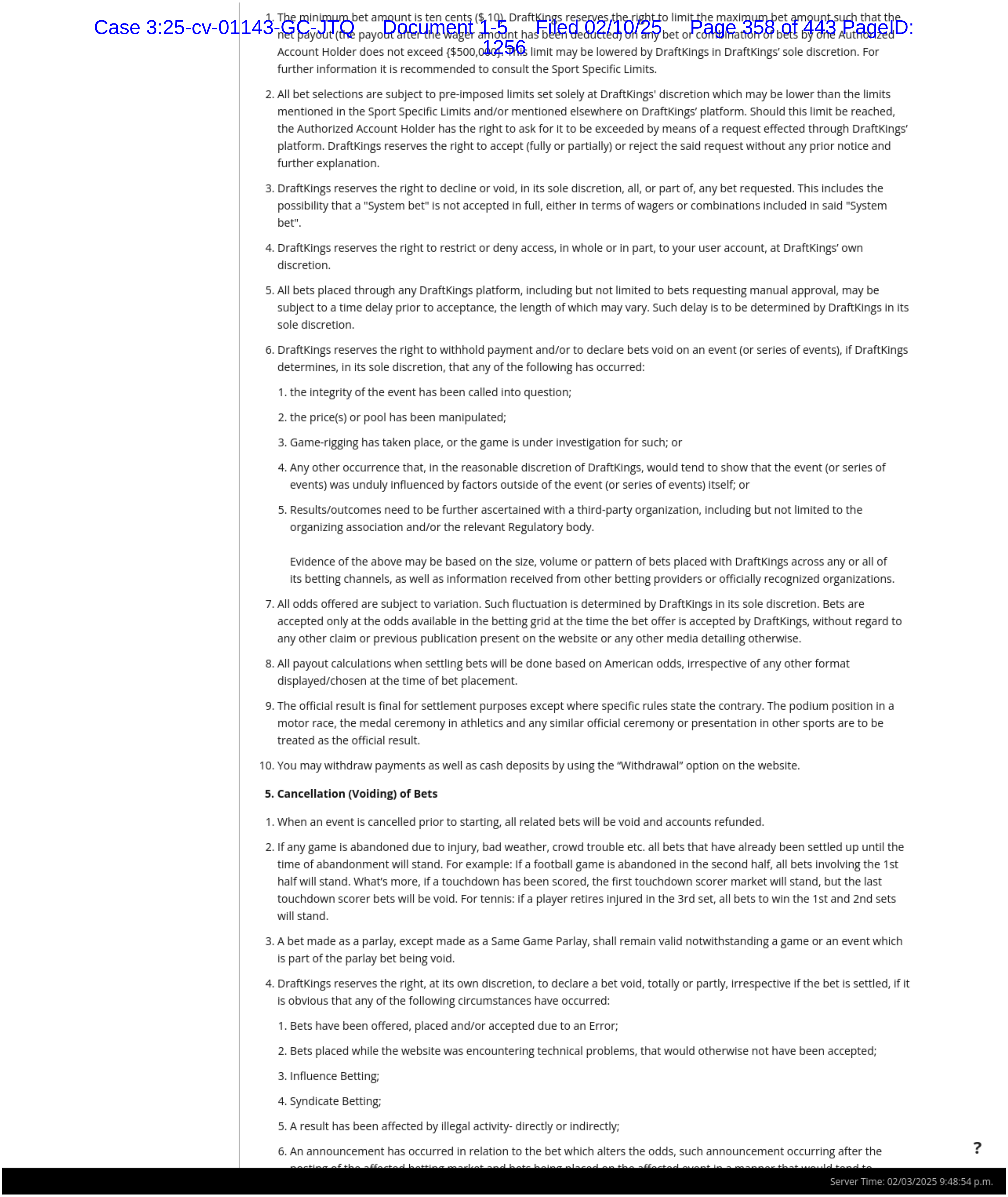
1. "Error" is a mistake, misprint, misinterpretation, mishearing, misreading, mistranslation, spelling mistake, technical hazard, registration error, transaction error, manifest error, force majeure and/or similar. Examples of errors include, but are not limited to:
   1. bets accepted during technical problems that would otherwise not have been accepted;
   2. bets placed on events/offers that have already been decided;
   3. bets on odds containing incorrect participants;
   4. bets placed at odds that are materially different from those available in the general market at the time the bet was placed;
   5. bets offered at odds which reflect an incorrect score and/or situation; or
   6. odds being clearly incorrect given the chance of the event occurring at the time the bet was placed.
2. "Pushes” are when wagers are refunded due to a tied outcome. References to “Push Rules” indicate that, for the associated market, ties will result in a refund of the wager amount as opposed to a win or loss.
3. "Influence Betting" is an act, prohibited by DraftKings, where an Authorized Account Holder, or parties acting in association with an Authorized Account Holder, can influence the outcome of a game or an event - directly or indirectly.
4. "Syndicate Betting" is an act, prohibited by DraftKings where Authorized Account Holders act together to place a series of bets on the same event or competition.
5. Where there is evidence of Authorized Account Holders acting together in this manner DraftKings reserves the right to make the relevant bets void and/or withhold payment of returns pending the outcome of subsequent investigations.

3. Bet Acceptance

1. A bet is not valid until it is validated and shows in the Authorized Account Holder’s bet history. In cases of uncertainty about the validity of a bet, the Authorized Account Holder is requested to check his/her open (pending) bets, or contact Customer Service.
2. Unless accepted in Error, once accepted, a bet will remain valid and cannot be withdrawn by you. It is the responsibility of the Authorized Account Holder to ensure details of the bets placed are correct. Under no circumstance will DraftKings accept any responsibility for any mistakes (perceived and actual), deriving from either Errors or any other reason, such as, but not limited to, incorrect listing of the odds/betting objects.
3. Should a dispute arise about the acceptance (or lack thereof) of any transaction in the Authorized Account Holder's account, the transaction log database will be the ultimate authority in deciding such matters.

4. Betting and Payout Limitations

1. The minimum bet amount is ten cents ($.10). DraftKings reserves the right to limit the maximum bet amount such that the net payout (the payout after the wager amount has been deducted) on any bet or combination of bets by one Authorized Account Holder does not exceed {$500,000}. This limit may be lowered by DraftKings in DraftKings’ sole discretion. For further information it is recommended to consult the Sport Specific Limits.



1. The minimum bet amount is ten cents ($.10). DraftKings reserves the right to limit the maximum bet amount such that the net payout (the payout after the wager amount has been deducted) on any bet or combination of bets by one Authorized Account Holder does not exceed {$500,000}. This limit may be lowered by DraftKings in DraftKings' sole discretion. For further information it is recommended to consult the Sport Specific Limits.
2. All bet selections are subject to pre-imposed limits set solely at DraftKings' discretion which may be lower than the limits mentioned in the Sport Specific Limits and/or mentioned elsewhere on DraftKings' platform. Should this limit be reached, the Authorized Account Holder has the right to ask for it to be exceeded by means of a request effected through DraftKings' platform. DraftKings reserves the right to accept (fully or partially) or reject the said request without any prior notice and further explanation.
3. DraftKings reserves the right to decline or void, in its sole discretion, all, or part of, any bet requested. This includes the possibility that a "System bet" is not accepted in full, either in terms of wagers or combinations included in said "System bet".
4. DraftKings reserves the right to restrict or deny access, in whole or in part, to your user account, at DraftKings' own discretion.
5. All bets placed through any DraftKings platform, including but not limited to bets requesting manual approval, may be subject to a time delay prior to acceptance, the length of which may vary. Such delay is to be determined by DraftKings in its sole discretion.
6. DraftKings reserves the right to withhold payment and/or to declare bets void on an event (or series of events), if DraftKings determines, in its sole discretion, that any of the following has occurred:
    1. the integrity of the event has been called into question;
    2. the price(s) or pool has been manipulated;
    3. Game-rigging has taken place, or the game is under investigation for such; or
    4. Any other occurrence that, in the reasonable discretion of DraftKings, would tend to show that the event (or series of events) was unduly influenced by factors outside of the event (or series of events) itself; or
    5. Results/outcomes need to be further ascertained with a third-party organization, including but not limited to the organizing association and/or the relevant Regulatory body.

       Evidence of the above may be based on the size, volume or pattern of bets placed with DraftKings across any or all of its betting channels, as well as information received from other betting providers or officially recognized organizations.
7. All odds offered are subject to variation. Such fluctuation is determined by DraftKings in its sole discretion. Bets are accepted only at the odds available in the betting grid at the time the bet offer is accepted by DraftKings, without regard to any other claim or previous publication present on the website or any other media detailing otherwise.
8. All payout calculations when settling bets will be done based on American odds, irrespective of any other format displayed/chosen at the time of bet placement.
9. The official result is final for settlement purposes except where specific rules state the contrary. The podium position in a motor race, the medal ceremony in athletics and any similar official ceremony or presentation in other sports are to be treated as the official result.
10. You may withdraw payments as well as cash deposits by using the "Withdrawal" option on the website.

5. Cancellation (Voiding) of Bets

1. When an event is cancelled prior to starting, all related bets will be void and accounts refunded.
2. If any game is abandoned due to injury, bad weather, crowd trouble etc. all bets that have already been settled up until the time of abandonment will stand. For example: If a football game is abandoned in the second half, all bets involving the 1st half will stand. What's more, if a touchdown has been scored, the first touchdown scorer market will stand, but the last touchdown scorer bets will be void. For tennis: if a player retires injured in the 3rd set, all bets to win the 1st and 2nd sets will stand.
3. A bet made as a parlay, except made as a Same Game Parlay, shall remain valid notwithstanding a game or an event which is part of the parlay bet being void.
4. DraftKings reserves the right, at its own discretion, to declare a bet void, totally or partly, irrespective if the bet is settled, if it is obvious that any of the following circumstances have occurred:
    1. Bets have been offered, placed and/or accepted due to an Error;
    2. Bets placed while the website was encountering technical problems, that would otherwise not have been accepted;
    3. Influence Betting;
    4. Syndicate Betting;
    5. A result has been affected by illegal activity- directly or indirectly;
    6. An announcement has occurred in relation to the bet which alters the odds, such announcement occurring after the posting of the affected betting market and bets being placed on the affected event in a manner that would tend to

?

Server Time: 02/03/2025 9:48:54 p.m.

Document title: General Rules | DraftKings Sportsbook
Capture URL: https://sportsbook.draftkings.com/help/general-betting-rules/general-rules
Capture timestamp (UTC): Mon, 03 Feb 2025 21:49:08 GMT

  3. Influence Betting;
  4. Syndicate Betting;
  5. A result has been affected by illegal activity directly or indirectly;
  6. An announcement has occurred in relation to the bet which alters the odds, such announcement occurring after the posting of the affected betting market and bets being placed on the affected event in a manner that would tend to show that the announcement affected the way that end users chose to bet.
  7. When a Customer places multiple copies of the same bet or places a number of bets that contain the same single selection. When this occurs, all bets may be voided apart from the first bet struck. An example would be where one particular selection is repeatedly included in multiple bets involving other short-priced selections; or
  8. Where there is evidence of a series of bets each containing the same (or very similar) selection(s) having been placed by the same individual or syndicate of individuals.

5. DraftKings reserves the right to void any bet that may have been accepted when the account did not have sufficient funds to cover the bet. If an account has insufficient funds as a result of a deposit that has been cancelled by the payment processing party, DraftKings reserves the right to void any bet that may have been accepted retroactively.
6. A bet made as a parlay, except made as a Same Game Parlay, shall never include two or more offers where the outcomes of which might turn out to be related (e.g. Team X to become champions and Player Y to be Top Goal Scorer in the same league). Although DraftKings takes reasonable steps to prevent such possibilities, in the eventuality that this would happen, DraftKings reserves the right, solely at its own discretion, to declare void all parts of the accumulative/parlay bet which include the correlated outcomes.
7. Bets can be voided regardless of whether the event has been settled or not.
8. Furthermore, all bets placed (and/or accepted) in the following circumstances will be declared void:
   1. Pre-game betting
      1. Betting effected after the event has started; or
      2. Betting effected after a related event was underway and where conditions could have been altered in a direct and indisputable way.
   2. Live betting
      1. Betting effected at incorrect price due to delayed or failing 'Live' coverage;
      2. Betting effected on particular offers after these have occurred, or else after an event which could normally be deemed as leading to the outcome has happened or is happening (e.g. bets placed on offers such as Total Goals Scored or Next Goal while a penalty is being taken, or has been awarded); or
      3. Betting effected on odds which represented a different score than the actual score

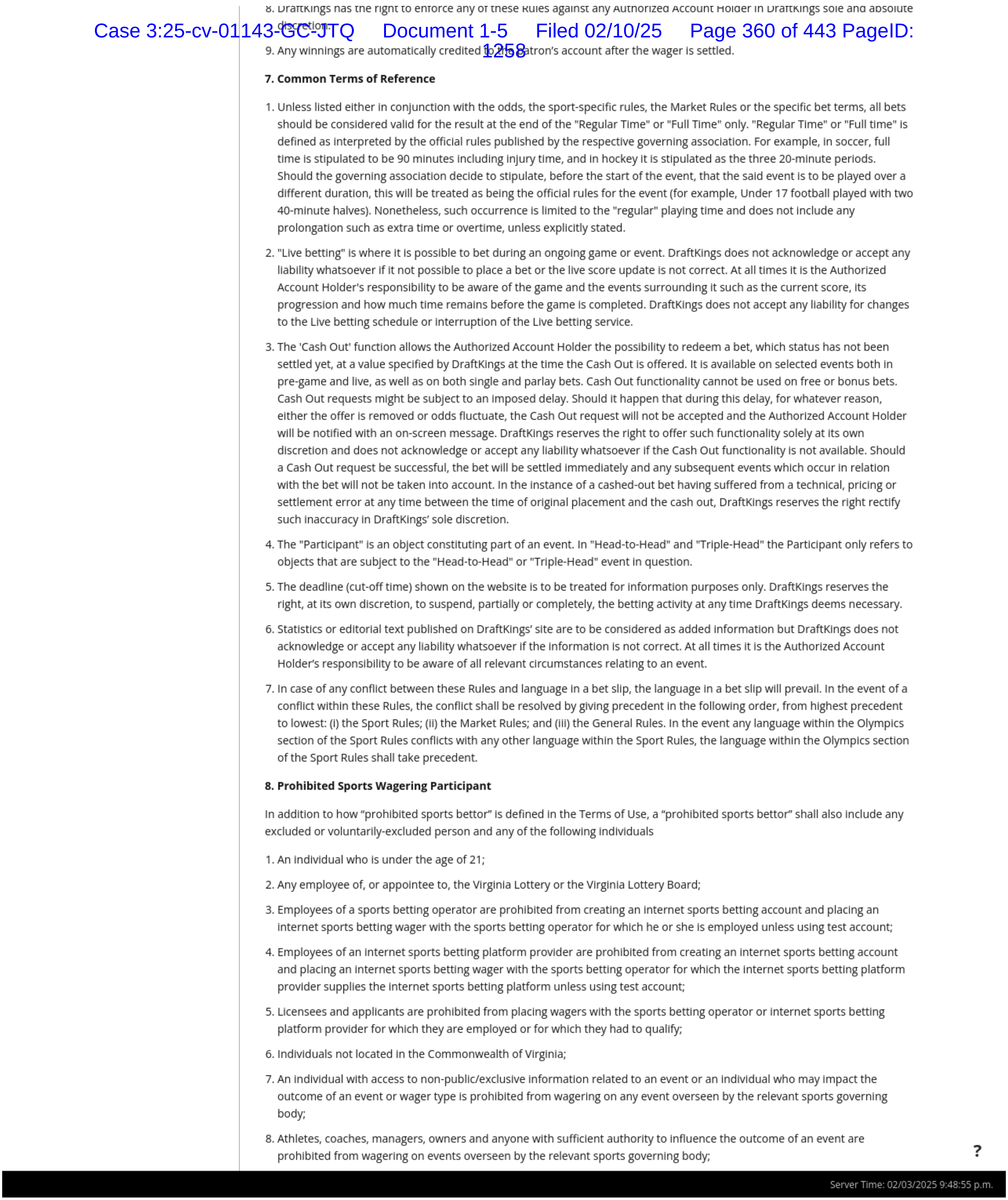
6. Disclaimer and Priority

1. DraftKings reserves the right, at its own discretion, to adjust a payout credited to an Authorized Account Holder's balance if it is obvious that the payout has been credited to the Account due to an Error.
2. In order to adjust any inaccuracy in the Authorized Account Holder's balance following amounts credited due to an Error, DraftKings reserves the right to take any necessary action, without prior notice and within reasonable limits, to adjust the Authorized Account Holder's balance through the reversal, amendment or cancellation, of any subsequent transaction on the Authorized Account Holder's account.
3. If you have a complaint regarding the DraftKings Sportsbook, please contact us directly at sportsbook@draftkings.com, and we will notify you of the disposition of any complaint within ten (10) calendar days of DraftKings receipt of your complaint. To the extent that any complaints you submit are not resolved to your satisfaction, you may submit the complaint to the Virginia Lottery in accordance with the Virginia Lottery's Sports Rules and Regulations. You consent to have DraftKings monitor and record any communications you have with DraftKings.
4. DraftKings reserves the right to suspend access to your account while any complaints are being resolved.
5. These Rules are applicable to all transactions with DraftKings Sportsbook and may be supplemented at any time in DraftKings' sole discretion. In the event that any of these Rules conflict with the Terms, the Terms shall govern.
6. In cases where it deems that these Rules are inconclusive, DraftKings reserves the right, according to its own discretion, to settle offers on an individual basis on the basis of equity, attaining itself to generally accepted betting norms, customs, and definitions.
7. Any data provided or accessible in, from or related to the Sportsbook may be used by the Authorized Account Holder for private, non-commercial use only and any use or attempted use of such data for commercial purposes is strictly prohibited.
8. DraftKings has the right to enforce any of these Rules against any Authorized Account Holder in DraftKings sole and absolute discretion.
9. Any winnings are automatically credited to the patron's account after the wager is settled.

7. Common Terms of Reference

?

Server Time: 02/03/2025 9:48:55 p.m.



8. DraftKings has the right to enforce any of these Rules against any Authorized Account Holder in DraftKings sole and absolute discretion.
9. Any winnings are automatically credited to the patron's account after the wager is settled.

7. Common Terms of Reference

1. Unless listed either in conjunction with the odds, the sport-specific rules, the Market Rules or the specific bet terms, all bets should be considered valid for the result at the end of the "Regular Time" or "Full Time" only. "Regular Time" or "Full time" is defined as interpreted by the official rules published by the respective governing association. For example, in soccer, full time is stipulated to be 90 minutes including injury time, and in hockey it is stipulated as the three 20-minute periods. Should the governing association decide to stipulate, before the start of the event, that the said event is to be played over a different duration, this will be treated as being the official rules for the event (for example, Under 17 football played with two 40-minute halves). Nonetheless, such occurrence is limited to the "regular" playing time and does not include any prolongation such as extra time or overtime, unless explicitly stated.
2. "Live betting" is where it is possible to bet during an ongoing game or event. DraftKings does not acknowledge or accept any liability whatsoever if it not possible to place a bet or the live score update is not correct. At all times it is the Authorized Account Holder's responsibility to be aware of the game and the events surrounding it such as the current score, its progression and how much time remains before the game is completed. DraftKings does not accept any liability for changes to the Live betting schedule or interruption of the Live betting service.
3. The 'Cash Out' function allows the Authorized Account Holder the possibility to redeem a bet, which status has not been settled yet, at a value specified by DraftKings at the time the Cash Out is offered. It is available on selected events both in pre-game and live, as well as on both single and parlay bets. Cash Out functionality cannot be used on free or bonus bets. Cash Out requests might be subject to an imposed delay. Should it happen that during this delay, for whatever reason, either the offer is removed or odds fluctuate, the Cash Out request will not be accepted and the Authorized Account Holder will be notified with an on-screen message. DraftKings reserves the right to offer such functionality solely at its own discretion and does not acknowledge or accept any liability whatsoever if the Cash Out functionality is not available. Should a Cash Out request be successful, the bet will be settled immediately and any subsequent events which occur in relation with the bet will not be taken into account. In the instance of a cashed-out bet having suffered from a technical, pricing or settlement error at any time between the time of original placement and the cash out, DraftKings reserves the right rectify such inaccuracy in DraftKings' sole discretion.
4. The "Participant" is an object constituting part of an event. In "Head-to-Head" and "Triple-Head" the Participant only refers to objects that are subject to the "Head-to-Head" or "Triple-Head" event in question.
5. The deadline (cut-off time) shown on the website is to be treated for information purposes only. DraftKings reserves the right, at its own discretion, to suspend, partially or completely, the betting activity at any time DraftKings deems necessary.
6. Statistics or editorial text published on DraftKings' site are to be considered as added information but DraftKings does not acknowledge or accept any liability whatsoever if the information is not correct. At all times it is the Authorized Account Holder's responsibility to be aware of all relevant circumstances relating to an event.
7. In case of any conflict between these Rules and language in a bet slip, the language in a bet slip will prevail. In the event of a conflict within these Rules, the conflict shall be resolved by giving precedent in the following order, from highest precedent to lowest: (i) the Sport Rules; (ii) the Market Rules; and (iii) the General Rules. In the event any language within the Olympics section of the Sport Rules conflicts with any other language within the Sport Rules, the language within the Olympics section of the Sport Rules shall take precedent.

8. Prohibited Sports Wagering Participant

In addition to how "prohibited sports bettor" is defined in the Terms of Use, a "prohibited sports bettor" shall also include any excluded or voluntarily-excluded person and any of the following individuals

1. An individual who is under the age of 21;
2. Any employee of, or appointee to, the Virginia Lottery or the Virginia Lottery Board;
3. Employees of a sports betting operator are prohibited from creating an internet sports betting account and placing an internet sports betting wager with the sports betting operator for which he or she is employed unless using test account;
4. Employees of an internet sports betting platform provider are prohibited from creating an internet sports betting account and placing an internet sports betting wager with the sports betting operator for which the internet sports betting platform provider supplies the internet sports betting platform unless using test account;
5. Licensees and applicants are prohibited from placing wagers with the sports betting operator or internet sports betting platform provider for which they are employed or for which they had to qualify;
6. Individuals not located in the Commonwealth of Virginia;
7. An individual with access to non-public/exclusive information related to an event or an individual who may impact the outcome of an event or wager type is prohibited from wagering on any event overseen by the relevant sports governing body;
8. Athletes, coaches, managers, owners and anyone with sufficient authority to influence the outcome of an event are prohibited from wagering on events overseen by the relevant sports governing body;

Server Time: 02/03/2025 9:48:55 p.m.

7. An individual with access to non-public confidential information related to an event or an individual who may impact the outcome of an event or wager type is prohibited from wagering on any event overseen by the relevant sports governing body;
8. Athletes, coaches, managers, owners and anyone with sufficient authority to influence the outcome of an event are prohibited from wagering on events overseen by the relevant sports governing body;
9. Employees of a sports governing body or its members teams are prohibited from wagering on any event overseen by the relevant sports governing body;
10. Owners of a sports governing body or member team are prohibited from wagering on any event overseen by the sports governing body or any event in which a member team of that sports governing body participates;
11. An individual on a list provided by a team or sports governing body may be prohibited from wagering on any event overseen by the relevant sports governing body;
12. Individuals wagering in violation of state or federal law;
13. An individual that has signed up for statewide self-exclusion; and
14. An individual whose participation may undermine the integrity of the wagering or the sports event or who is excluded from wagering for other good cause, including but not limited to, an attempt to place a wager as an agent or a proxy.

If any bets are placed by any prohibited sports wagering participants, such bets shall be cancelled and refunded.

9. Prohibited Sports Wagering Events And Wagers

Prohibited sports wagering events and wagers include:

1. Wagering on horse racing authorized by Chapter 29 (§ 59.1-364 et seq.) of Title 59.1
2. Any event played by individuals that are at the high school level or below unless the majority of participants in the event are 18 years of age or older.
3. Roulette, poker, blackjack, a card game, a dice game, or any other game or contest typically offered in a casino other than sports betting.
4. A fantasy contest.
5. Any proposition wager placed on any type of possible injury, unsportsmanlike conduct, or any other officiating call
6. Any proposition bet on college sports.
7. Any bet on college sports for colleges/universities located in the Commonwealth of Virginia.
8. Any event category, event, or wager type prohibited by the Lottery or by state or federal law.

10. Methods of funding a wager

Sports Wagers may be funded through multiple options, including, without limitation, customer deposits and site/promotional credits. Deposits can be made through Debit Cards, Online Banking, PayPal, Play+ Cards, Bank Wire Transfers, Cash at Retail, and approved Gift Cards, and may include any other method approved by the Virginia Lottery.

11. Dead Heat Reduction

1. In the event a bet sees two or more competitors tied for the same winning position (according to the terms of the bet), the "Dead Heat Reduction ("DHR") rules" apply.
2. A "Dead Heat Reduction" is calculated by dividing the odds proportionally among the number of winners for a particular position (i.e. finishing place) in the event. For example, in a two-way tie aka 'Dead Heat', your return would be half of what was originally projected in the bet slip at the time of bet placement.
    1. Example (all calculations should use decimal odds):
        1. You bet $50 wager on a golfer to have the best score among a group of 3 at +150 (2.5) odds.
        2. The event ends with two golfers tying for the best score in the group.
        3. 2.5 (original odds) ÷ 2 (number of participants who tied) = 1.25 (DHR odds)
        4. $50 x 1.25 = $62.50 payout
3. A "Dead Heat Reduction" involving a bet with more than one finishing position (e.g. Top 10) is calculated as follows:
    1. Divide the number of participants who tied by the number of remaining finishing positions.
    2. Divide your original odds by the result of the calculation immediately above to find the new odds.
    3. Example (all calculations should use decimal odds):
        1. You bet $50 wager on a golfer to place in the Top 10 at +200 (3.0) odds.
        2. The event ends with the golfer tying for 8th place with 5 other golfers.
        3. 6 (tied participants) ÷ 3 (8th, 9th & 10th)) = 2.0
        4. 3.0 (original odds) ÷ 2.0 (new odds) = 1.5 (DHR odds)

?

Server Time: 02/03/2025 9:48:56 p.m.

18 years of age or older.

3. Roulette, poker, blackjack, a card game, a dice game, or any other game or contest typically offered in a casino other than sports betting.
4. A fantasy contest.
5. Any proposition wager placed on any type of possible injury, unsportsmanlike conduct, or any other officiating call
6. Any proposition bet on college sports.
7. Any bet on college sports for colleges/universities located in the Commonwealth of Virginia.
8. Any event category, event, or wager type prohibited by the Lottery or by state or federal law.

10. Methods of funding a wager

Sports Wagers may be funded through multiple options, including, without limitation, customer deposits and site/promotional credits. Deposits can be made through Debit Cards, Online Banking, PayPal, Play+ Cards, Bank Wire Transfers, Cash at Retail, and approved Gift Cards, and may include any other method approved by the Virginia Lottery.

11. Dead Heat Reduction

1. In the event a bet sees two or more competitors tied for the same winning position (according to the terms of the bet), the "Dead Heat Reduction ("DHR") rules" apply.
2. A "Dead Heat Reduction" is calculated by dividing the odds proportionally among the number of winners for a particular position (i.e. finishing place) in the event. For example, in a two-way tie aka 'Dead Heat', your return would be half of what was originally projected in the bet slip at the time of bet placement.
   1. Example (all calculations should use decimal odds):
      1. You bet $50 wager on a golfer to have the best score among a group of 3 at +150 (2.5) odds.
      2. The event ends with two golfers tying for the best score in the group.
      3. 2.5 (original odds) ÷ 2 (number of participants who tied) = 1.25 (DHR odds)
      4. $50 x 1.25 = $62.50 payout
3. A "Dead Heat Reduction" involving a bet with more than one finishing position (e.g. Top 10) is calculated as follows:
   1. Divide the number of participants who tied by the number of remaining finishing positions.
   2. Divide your original odds by the result of the calculation immediately above to find the new odds.
   3. Example (all calculations should use decimal odds):
      1. You bet $50 wager on a golfer to place in the Top 10 at +200 (3.0) odds.
      2. The event ends with the golfer tying for 8th place with 5 other golfers.
      3. 6 (tied participants) ÷ 3 (8th, 9th & 10th)) = 2.0
      4. 3.0 (original odds) ÷ 2.0 (new odds) = 1.5 (DHR odds)
      5. $50 x 1.5 = $75 payout

DraftKings Inc.
Boston, MA

COMPANY

- About DraftKings
- Careers
- Mobile Apps
- Affiliates
- Refer a Friend
- Get Help
- Terms Of Use
- Privacy Policy
- Responsible Gaming
- Retail Sportsbook Locations
- Accessibility
- DKNetwork Sports News
- Do Not Sell or Share My Personal Information

SPORTSBOOK ODDS

- NFL Odds
- NBA Odds
- MLB Odds
- UFC Odds
- NHL Odds
- PGA Odds
- Soccer Odds
- Tennis Odds
- College Basketball Odds
- College Football Odds
- Live Odds
- Super Bowl Odds
- NFL Playoffs Odds

BETTING GUIDES

- How To Bet
- How to Bet on Football
- How to Bet on Basketball
- How to Bet on UFC
- How to Bet on Baseball
- How to Bet on Hockey
- How to Bet on Golf
- How to Bet on Soccer
- How to Bet on Tennis
- How to Bet on NASCAR

LOCATIONS

US Office
222 Berkeley Street
Boston, MA
02116

New Jersey Office
PO Box 399
Hoboken, NJ
07030

UK Office
15 Ingestre Place
Suite 265
Soho, London
W1F OJH

Must be 21+ and in Virginia to place wagers. If you or someone you know has a gambling problem, call the Virginia Problem Gambling Helpline at 888-532-3500 or visit Virginia Council on Problem Gambling. Subject to regulatory licensing requirements.

21+ RG

Server Time: 02/03/2025 9:49:01 p.m.

Exhibit 62

Intro — Sports Intelligence @ DraftKings

Robin Mohseni · Follow
Published in DraftKings Engineering · 7 min read · Jul 14, 2023

28 1

Co-written by Ian Dorward, Prashant Singh and Robin Mohseni

At DraftKings, we have a proven track record of providing our customers with a world-class sports betting experience. We are proud to present our newly established Sports Intelligence team, aimed at redefining the sports betting landscape through innovation, expertise and cutting-edge technology. With a focus on delivering the best sports betting content in the industry, our dynamic team comprises two major units: Sports Data Engineering and Sports Data Science.

Sports Data Engineering: Laying the Foundation for Success

The Sports Data Engineering team plays a pivotal role within DraftKings' Sports Intelligence division. This highly skilled team is responsible for creating and curating best-in-class data assets that serve as the building blocks for the advanced modelling capabilities of the Sports Data Science team.

Additionally, the Sports Data Engineering team excels in developing robust model replay features. This powerful tool enables the evaluation and optimization of sports models by simulating past events, thus assessing algorithm efficiency under varying conditions. By continuously improving DraftKings' sports betting content, these endeavours enhance the overall customer experience.

Sports Data Science: Driving Innovation in Sports Modelling

The creation of the Sports Data Science team at DraftKings can be traced back to 2019, when it was originally formed by SBTech before the acquisition by DraftKings Inc. From the beginning, the team recognized the immense opportunities to revolutionise sports modelling capabilities across various sports, starting with the fundamentals. Initially focused on the US betting market, the team quickly embarked on rebuilding major sports, particularly football and basketball, and expanded to cover all major sports.

sports, starting with the fundamentals. Initially focused on the US betting market, the team quickly embarked on building major sports, particularly football and basketball, and expanded to cover all major sports.

To provide a behind-the-scenes look at how sports data science is conducted at DraftKings, our team has been working on a series of articles. We will start with a short introduction discussing common techniques used and what we believe sets us apart from the competition.

Understanding "Sports Data Science"

Traditionally, teams like ours in the industry were referred to as 'Quantitative Analysis', 'Quantitative Development' or simply 'Quants', while 'Data Science' was focused on customer intelligence (recommender systems, customer segmentation and experimentation). The majority of Quants come from a very strong maths background, working with Trading desks on pricing models and various trading tools. In our experience, these teams used to build models in Excel, VBA, R and handover these to other engineering teams to run the models at scale. Often some of these legacy models had an Excel front-end (commonly referred to as RoboTraders). More recently, Quant teams have adopted more common object-oriented programming languages such as C# or Java to deliver more end-to-end but this Modelling-to-Engineering hand off for model productionisation still exists to a high degree.

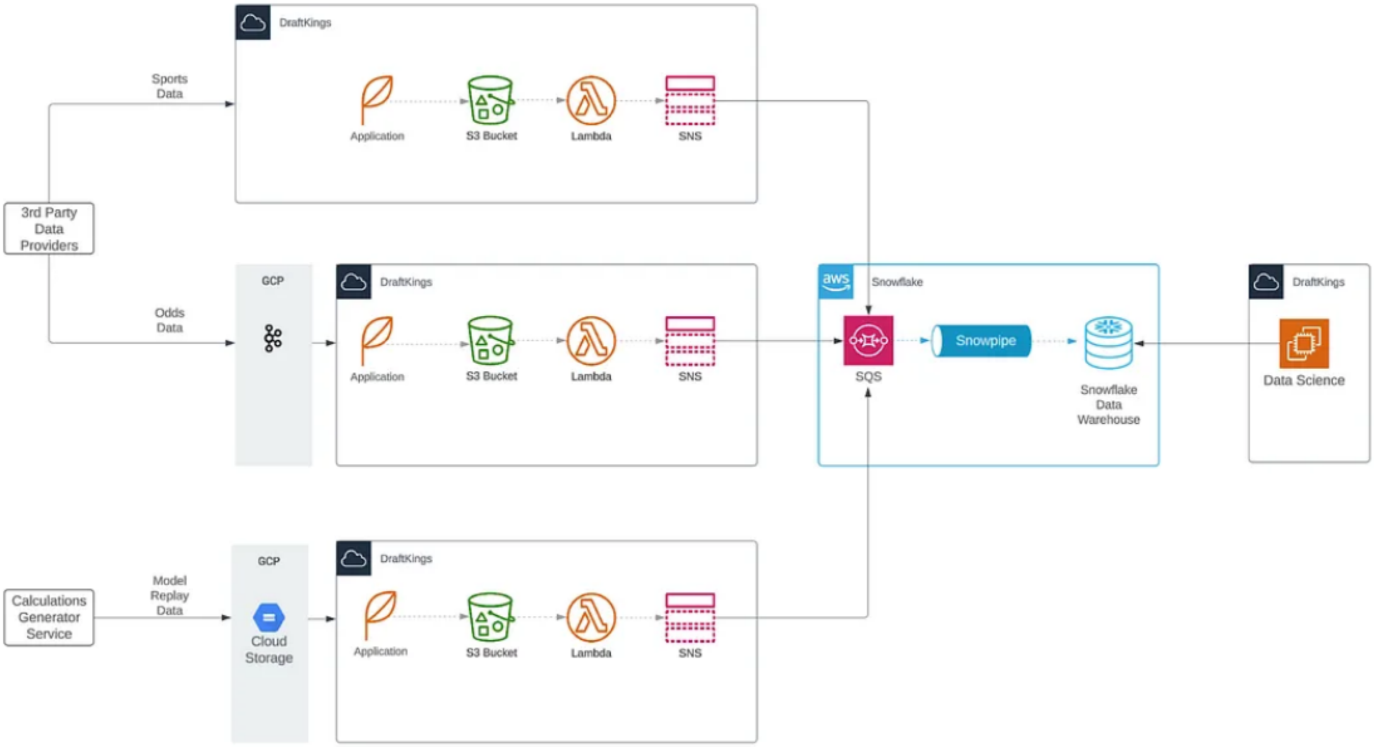
At DraftKings, we consider ourselves to be a Data Science function. While we possess strong mathematical skills and collaborate closely with trading stakeholders, we have made a conscious effort to prioritise being a data-centric engineering team leveraging the latest machine learning techniques for sports modelling, capable of shipping machine learning products to Production.

The UK and Ireland are hotbeds for talent given the mature nature of the sports betting industry in Europe compared to the US, so the Sports Data Science team is based in London and Dublin. We have hired team members from a wide variety of backgrounds whilst also understanding the importance of domain expertise in this area, amassing decades of sports modelling experience within the group.

Photo from a team Golf Day in 2022

Engineering Technology Stack: Powering Innovation through C#, SQL, Kafka, AWS and Snowflake

Our Engineering team works with a wide ranging technology stack. We leverage programming languages such as C# and SQL to ingest and transform raw data and we use the open-source data streaming tool Kafka for ingesting real-time data feeds. Our cloud infrastructure is deployed on AWS with robust security measures in place and we persist the data in the Snowflake Data Warehouse.



Data Architecture

Data Science Technology Stack: Powering Innovation through Python and Kubernetes

In terms of our technology, the data science team works almost exclusively in Python. While we conduct extensive research and development in notebooks, both locally and on Databricks, our models are deployed as containerized microservices on Kubernetes. This approach ensures stability and resilience, which are crucial for high-frequency, real-time solutions like ours. Despite Python's reputation for slower execution, we have worked on solutions over the years to meet SLAs for calculation speed.

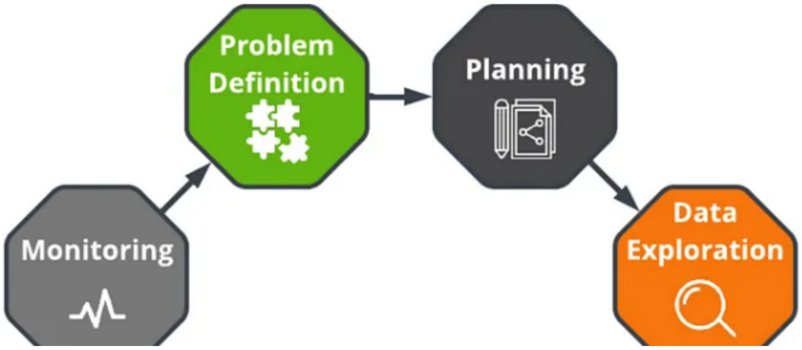
Building Sports Models at DraftKings: The Monte-Carlo Approach

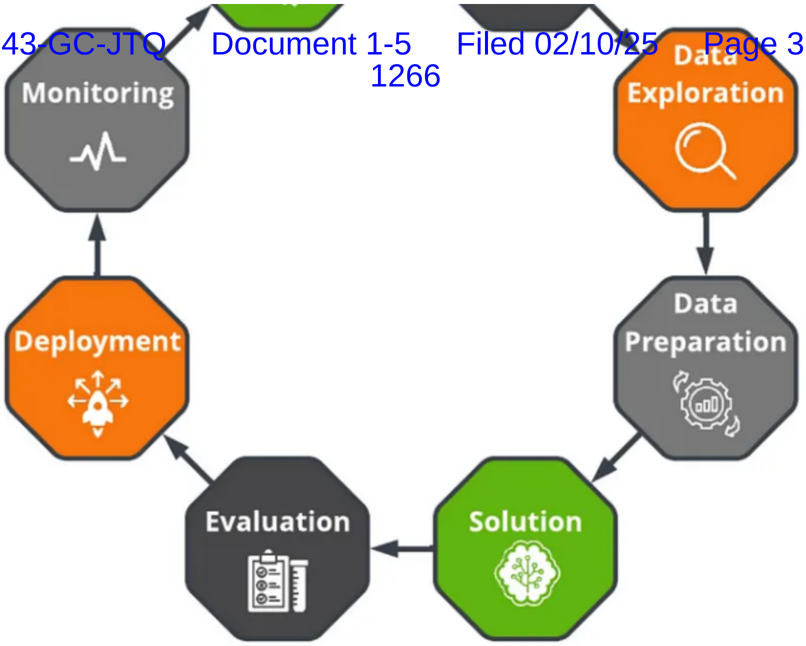
At DraftKings, the majority of our core models rely on Monte-Carlo simulation engines. These engines simulate entire games as stochastic processes, where each state transition is predicted using various machine learning models. The chosen unit for each state transition varies based on the sport, aiming to provide the most accurate and extensible products for our customers, whilst minimising computational latency and complexity. Our stateless calculations enable easy debugging and historical game replays.

Embracing MLOps: Achieving Best Practices

We follow MLOps best practices and leverage our ML Platform to facilitate seamless model deployment. Our ML models reside in a cloud-based ML model registry, exposed to calculation engines via APIs. During the inference stage, real-time data feeds and trader inputs are integrated into the calculation engines to generate the odds that our customers see. We constantly iterate on the ML models and simulation engines, leveraging new data sources and new modelling techniques.

Sports Data Science SDLC: Constant Improvement





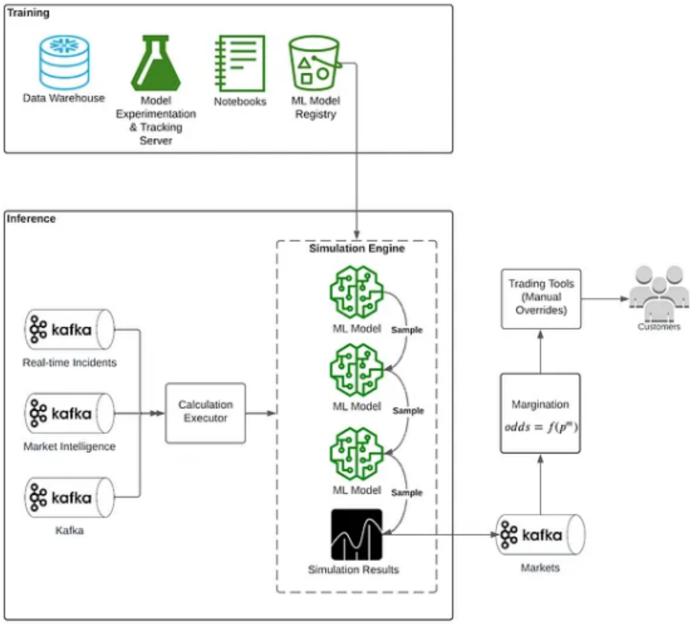
Sports Data Science Software Development Lifecycle

The high-level calculation engine lifecycle is as follows, and includes iterative loops at many levels. It combines many of the traditional aspects of a data science SDLC, whilst also building in domain-specific steps.

1. Work closely with Product and other stakeholders to clearly define the problem statement and requirements
2. Carry out exploratory data analysis and ideation using data curated by the Sports Data Engineering team in our best-in-class Sports Data Warehouse
3. Perform feature engineering, set up training pipelines and validate results
4. Encode sport logic into play-by-play or time based simulations
5. Integrate into our real-time streaming architecture
6. Use simulation results to calculate probability distributions and markets
7. Add margin to markets based upon configuration and margination algorithms
8. Propagate markets to trading tools for manual inputs or odds adjustments and risk management

The diagram below broadly illustrates the general flow for calculation inference for our models. In reality, the feedback loops between our Traders and the models are more complex.

and the models are more complex.



Simulation Flow

Integration with the Sportsbook: Real-Time Streaming Architecture

Our sports models, deployed on Kubernetes, offer HTTP endpoints for calculation execution. Simulation results are produced to Kafka topics, supporting calculations for both single markets and Same Game Parlays (SGPs). In future articles, we will delve into the intricacies of this architecture, including challenges such as odds jumping as a function of simulation variance.

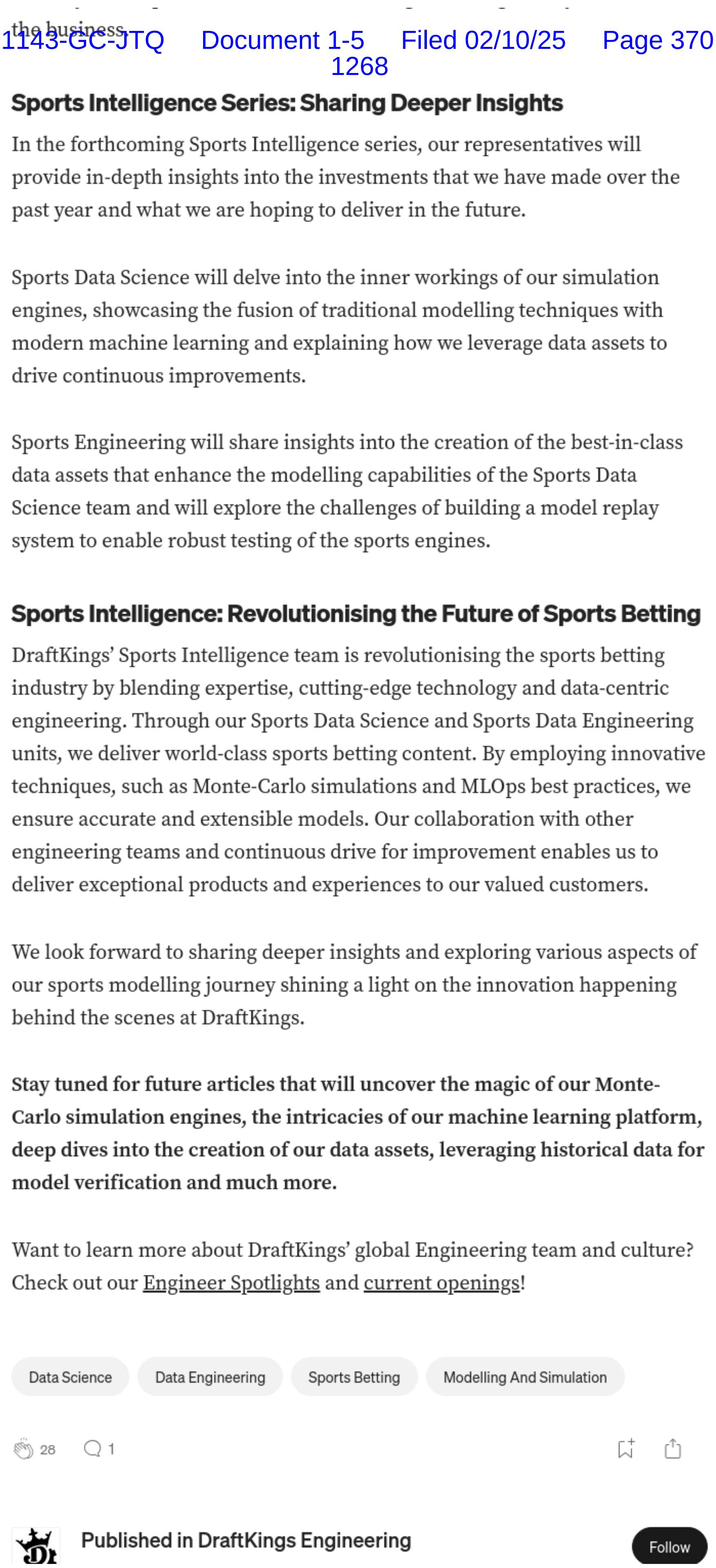
Collaboration and Innovation: Delivering Exceptional Products

To bring our complex products to Production, our team collaborates with other talented engineering teams at DraftKings, including Sports Data, Markets and Sports Platform. Recently, we introduced live SGPs for NFL and NBA, showing our commitment to innovation and customer satisfaction.

Furthermore, our team supports algorithms for features such as margination and cashout. As there is a limited need for training data and machine learning in these cases, and the implementations of these calls hit the millions — the algorithms are provided to other engineering teams in the form of Nuget packages that were converted from Python to C#. These have recently been updated and we are now using C bindings via Python across the business.

Sports Intelligence Series: Sharing Deeper Insights

In the forthcoming Sports Intelligence series, our representatives will



the business.

Sports Intelligence Series: Sharing Deeper Insights

In the forthcoming Sports Intelligence series, our representatives will provide in-depth insights into the investments that we have made over the past year and what we are hoping to deliver in the future.

Sports Data Science will delve into the inner workings of our simulation engines, showcasing the fusion of traditional modelling techniques with modern machine learning and explaining how we leverage data assets to drive continuous improvements.

Sports Engineering will share insights into the creation of the best-in-class data assets that enhance the modelling capabilities of the Sports Data Science team and will explore the challenges of building a model replay system to enable robust testing of the sports engines.

Sports Intelligence: Revolutionising the Future of Sports Betting

DraftKings' Sports Intelligence team is revolutionising the sports betting industry by blending expertise, cutting-edge technology and data-centric engineering. Through our Sports Data Science and Sports Data Engineering units, we deliver world-class sports betting content. By employing innovative techniques, such as Monte-Carlo simulations and MLOps best practices, we ensure accurate and extensible models. Our collaboration with other engineering teams and continuous drive for improvement enables us to deliver exceptional products and experiences to our valued customers.

We look forward to sharing deeper insights and exploring various aspects of our sports modelling journey shining a light on the innovation happening behind the scenes at DraftKings.

Stay tuned for future articles that will uncover the magic of our Monte-Carlo simulation engines, the intricacies of our machine learning platform, deep dives into the creation of our data assets, leveraging historical data for model verification and much more.

Want to learn more about DraftKings' global Engineering team and culture? Check out our Engineer Spotlights and current openings!

Published in DraftKings Engineering

Published in DraftKings Engineering

285 Followers · Last published Oct 2, 2024

DraftKings Engineering

Follow

Written by Robin Mohseni

30 Followers · 6 Following

Senior Director, Data Science

Follow

Responses (1)

What are your thoughts?

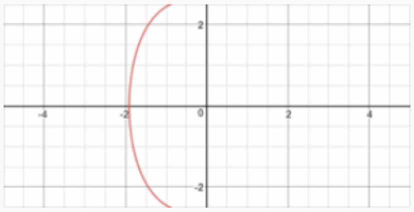
Respond

Hugo Lu
Oct 2, 2023

Bravo!

Reply

More from Robin Mohseni and DraftKings Engineering

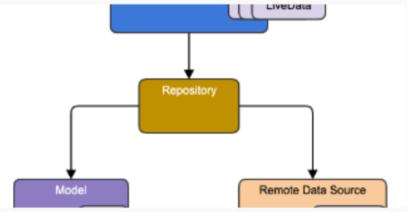


In DraftKings Engineering by Mehrad Kavian

Signature Malleability

Many smart contracts rely on the validity of a signature or whether it has been used befor...

Jun 20, 2023 30



In DraftKings Engineering by Mike DeMaso

Android, MVVM, and Repositories in the Real World

It is a tale as old as Android Development itself. How do we achieve a responsive user...

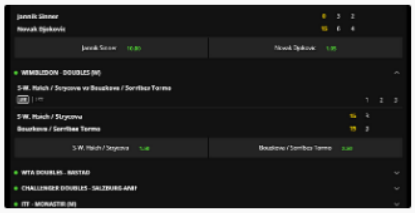
Nov 2, 2020 21

In DraftKings Engineering by Martin Chaov

Lessons Learned: WebSocketAPI at scale

Here at Draftkings Inc. we utilize the WebSocket API for a variety of use-cases...

May 5, 2023 42 1



In DraftKings Engineering by Ian Dorward

Building a Tennis Simulation

Tennis has become one of the most popular sports at DraftKings, ranking as the fourth...

Aug 17, 2023 137

See all from Robin Mohseni

See all from DraftKings Engineering

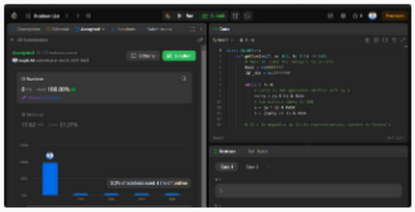
Recommended from Medium

In DataDrivenInvestor by Austin Starks

I used OpenAI's o1 model to develop a trading strategy. It is...

It literally took one try. I was shocked.

Sep 15, 2024 8.6K 217



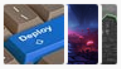
Aaqib Ali

Day-1, Day-2 LeetCode Challenge-365-Days.

I have started my journey to prepare for Google interviews and set a complete year-...

Jan 1

Lists

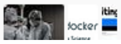


Predictive Modeling w/ Python

20 stories · 1812 saves

Practical Guides to Machine Learning

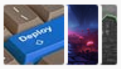
10 stories · 2185 saves



Coding & Development

ChatGPT prompts

Lists



Predictive Modeling w/ Python
20 stories · 1812 saves

Practical Guides to Machine Learning
10 stories · 2185 saves

Coding & Development
11 stories · 991 saves

ChatGPT prompts
51 stories · 2531 saves

Alberto Romero

DeepSeek Is Chinese But Its AI Models Are From Another Planet

OpenAI and the US are in deep trouble

Jan 22 5.8K 182

In Level Up Coding by Jacob Bennett

The 5 paid subscriptions I actually use in 2025 as a Staff Software...

Tools I use that are cheaper than Netflix

Jan 7 7.7K 171

Sofiane Trabelsi

How Chess Shaped My Journey as a Software Engineer

Growing up, I never imagined how much a board game could influence my career....

Jan 12 557 3

Jessica Stillman

Jeff Bezos Says the 1-Hour Rule Makes Him Smarter. New...

Jeff Bezos's morning routine has long included the one-hour rule. New...

Oct 30, 2024 21K 603

See more recommendations

Help Status About Careers Press Blog Privacy Terms Text to speech Teams

Exhibit 63

Apache Kafka as a centralized state management platform — Part 2

Nicola Atorino · Follow

Published in DraftKings Engineering · 13 min read · Jul 7, 2023

13

Kafka for centralized storage and state sharding

Apache Kafka, an open-source distributed streaming platform, is used by DraftKings (DK) for data management and streaming applications due to its high scalability, low latency, and fault-tolerant storage. Kafka allows for various use cases such as data streaming, and state management. DraftKings employs Kafka as a push-based mechanism to distribute data from a database to multiple services, reducing strain on the database and internal traffic, following several architectural patterns that can cover different use cases. In the previous article, we explored the capabilities of Apache Kafka for state management and described a pattern to leverage its use to offload database traffic. In this article, we're going to describe two additionally useful patterns that will solve some of the problems outlined in the previous cases and will extend our possibilities in terms of distributed processing.

The first part of this article can be found here : https://medium.com/draftkings-engineering/apache-kafka-as-a-centralized-state-management-platform-part-1-aa6d3fe4024b

Use Kafka as Centralized Store for Distributed Applications

This pattern is useful to distribute a stateful application. In these cases, a microservice would need to consume a stream of messages that sends updates for particular events, then maintain an aggregate of information for each event, in order to handle further processing. It's not unusual for this kind of stream to produce thousands of messages per second, so the partitioning capabilities of Kafka are extremely valuable to orchestrate the consumption. By implementing a set of consumers in the same consumer group, processing will be distributed correctly between the many instances of the application, and rebalancing in case of crashes, deployments, or change in the number of instances will be correctly handled by the broker and the consumer group coordinator out of the box.

Use cases example:

change in the number of instances will be correctly handled by the broker and the consumer group coordinator out of the box.

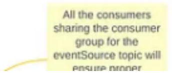
Use cases example:

- updates of probabilities for sport events that need to be compared to the previous values
- stock exchange price fluctuation — to generate an analysis of changes overtime or produce an average per day
- wallet management — maintain a log of transactions together with a real-time snapshot of the balance in order to accept/refuse new transactions processing
- fraud detection — keep track of behavioral patterns and detect anomalies over an established flow
- shared sessions in multiplayer games — keep a centralized leaderboard up to date by events coming from different producers

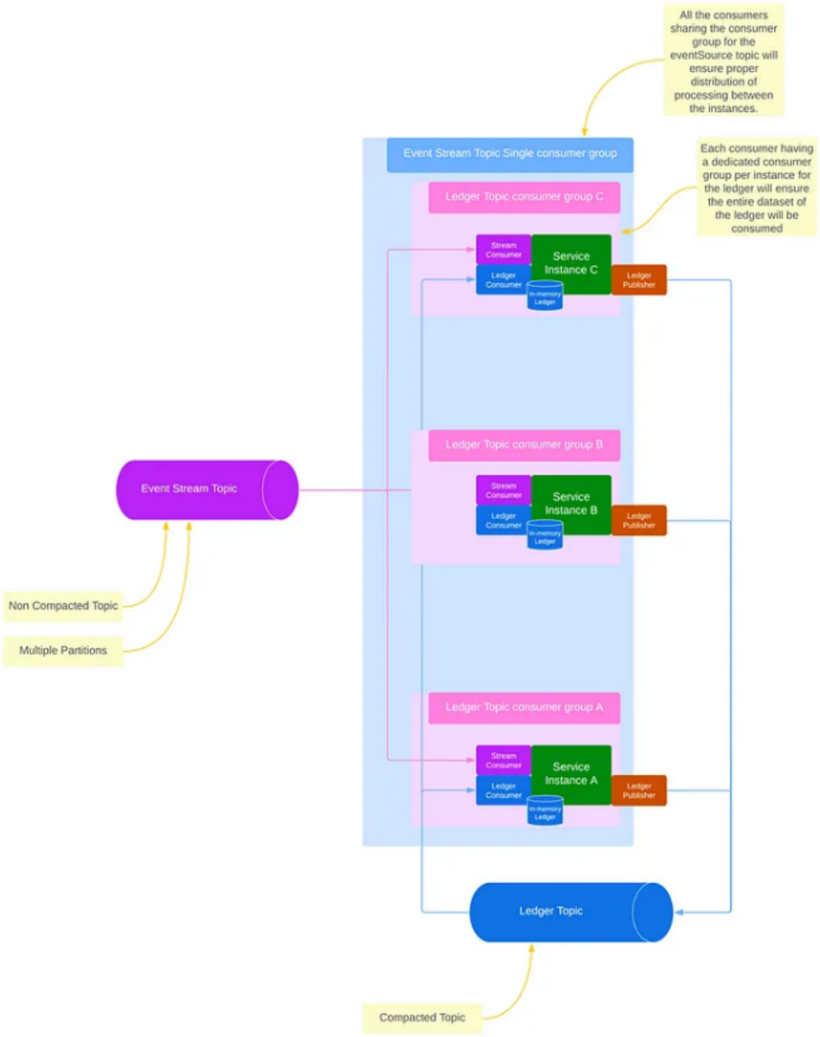
To handle state management, it is possible to implement a dedicated ledger topic where each instance of the service can read from and write to. The ledger topic will then act as a centralized, persistent store for the system, surviving service crashes and redeployments. To allow this, a ledger consumer can be implemented in the service, this time with a different consumer group per instance, to ensure that each instance has the entire state at its disposal, including everything that was published by any other replica. The service can then maintain the ledger in its own memory, populate it in full form at the startup, and receive updates when they are produced, this way achieving almost real-time synchronization between the various components.

In case of rebalancing, the stream topic will be evenly distributed following the new topology, and every consumer can then resume from the last committed offset for its own consumer group, with the guarantee that the ledger already contains all necessary information processed up until this point.

There is a short-circuit mechanism in place to avoid concurrency issues due to the fact that a published message has to be consumed by the instance that published it. This allows every instance, upon publishing the state update to the ledger, to populate its own internal memory. This means the published message does not have to wait the round-trip to the Kafka broker between publishing and re-consuming the same message that it produced in order to populate its own state. This guarantees protection against race conditions.


All the consumers sharing the consumer group for the eventSource topic will ensure proper

publishing and re-consuming the same message that it produced in order to populate its own state. This guarantees protection against race conditions.


All the consumers sharing the consumer group for the eventSource topic will ensure proper distribution of processing between the instances.

Each consumer having a dedicated consumer group per instance for the ledger will ensure the entire dataset of the ledger will be consumed

Design Pattern for a centralized ledger

Considerations

One thing to take into consideration when planning a solution like this is the actual coordination between stream processing and ledger processing. To achieve 100% data consistency, the offsets of the processing topic must be committed only after there is a guarantee from the broker that the ledger has been correctly updated. Otherwise, in case of crashes or ungraceful shutdown there is the risk of committing an offset — thus marking a stream message as consumed — before the ledger is updated with the result of the processing. This kind of synchronization, however, can be costly in terms of performance, so the decision to handle these edge cases explicitly instead of relying on a simpler solution has to be taken based on the criticality of the business case, the impact of potential data loss, and the possibility of this happening in a given infrastructure.

One other consideration to take is data retention. The TTL of the items in memory do not necessarily have to match the retention of the ledger topic relying on a simpler solution has to be taken based on the criticality of the business case, the impact of potential data loss, and the possibility of this happening in a given infrastructure.

One other consideration to take is data retention. The TTL of the items in memory do not necessarily have to match the retention of the ledger topic, however, is usually advised to do so in to have consistency between the two layers. An example of this is to set the same retention period between the items in memory and the Kafka ledger. Alternatively, the topic can be set to have no retention at all. This is useful in cases when the decision to remove an item from the ledger is taken explicitly by the application, that can send a deletion request when it deems a specific item not necessary anymore. In this case, the ledger will receive what it's called a 'tombstone' message, that will trigger removal of all messages with the same key upon compaction. Tombstone messages are consumed like any other message, and the service can react to this particular message type by removing the item from the memory state, thus keeping synchronization with the state of the ledger. It is important to make sure that memory purging is properly handled in the application, time-based or explicitly, otherwise memory leaks are guaranteed in the long run.

Use Cases

A pattern like this is very useful when the partitioning logic of the stream topic does not match the partitioning logic of the ledger. If this is the case, there is no guarantee that an instance processing a message from a specific partition may never require data that was retrieved and published by another instance. Another use case is when the complete state is needed for overall processing, like notifying the system with an atomic update that any of the events is in a particular state.

An example for the first case described above may be an application that maps external data to internal entities. In this case, the input topic may be partitioned based on the 3rd party event Id, while the output topic will need to be partitioned by the internal event Id. Of course, the Ids won't match in this case, so there is no way of knowing in which partition the published internal event will end up. When, instead, the state of an instance can be properly isolated from the state of the others, we could spice this pattern up a notch and introduce state sharding.

Use Kafka to Handle State Sharding

Let's now consider a slightly different use case when each instance of a distributed system does NOT need to handle state management outside of the partitions that have been assigned to it. In this situation, it is not necessary for each replica to consume the entire ledger topic, but instead, it will be enough to consume only the part of it that is usually tied to the partition that has been assigned to it for the stream.

the partitions that have been assigned to it. In this situation, it is not necessary for each replica to consume the entire ledger topic, but instead, it will be enough to consume only the part of it that is usually tied to the partition that has been assigned to it for the stream.

In this case, the configuration of the ledger topic must match the configuration of the stream topic, in terms of number of partitions and partitioning key. This way, every event that is streamed from partition 1 will store its current state in partition 1 of the ledger, and so forth. With this approach, each instance does not have to maintain the entire ledger in memory, only the shard related to the partitions it has been assigned to. This has obvious advantages in terms of scalability and resource management when it comes to high volume of items to track.

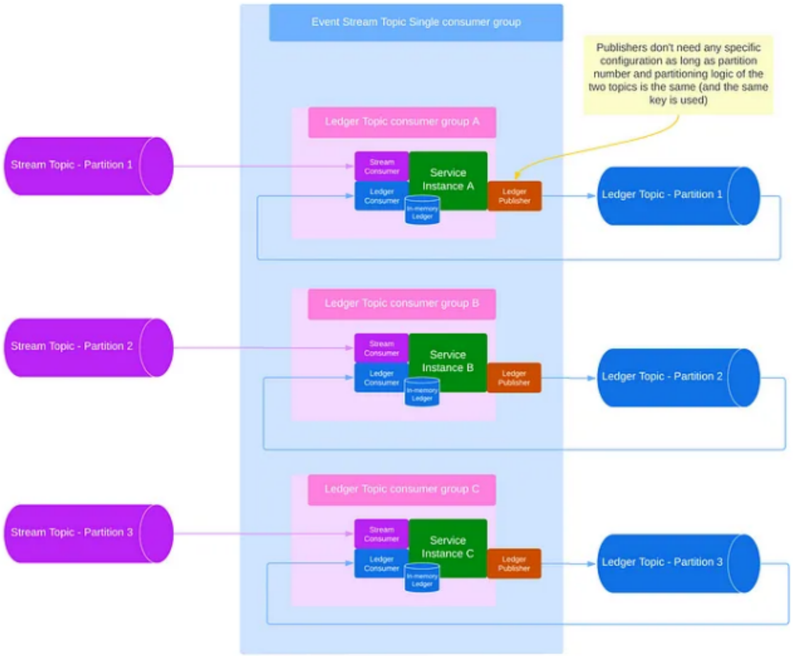
However, there is a complication. When rebalancing, partition reassignment is automatic and the default mechanism of the broker is to assign them based on load. Therefore, predicting what partitions would be assigned to a specific consumer can be complicated. The consumption of the ledger must follow the assignment of the stream topic partitions, with the instance explicitly consuming a specific set of ledger partitions, based on the stream partitions that were assigned to it by the consumer group coordinator for the stream topic.

When a group rebalancing for the stream topic is triggered:

- The consumer will stop consumption from the currently assigned partitions. This ensures no concurrency issues can happen during the rebalancing process
- The consumer will receive a notification that the currently assigned partitions have been removed. This can be used as a trigger to stop consumption of the ledger partitions that have been removed and remove every item in the state coming from the removed partitions
- Once all consumers have confirmed that they have handled the PartitionRemoved event correctly, the consumer will receive a notification that a new set of partitions have been assigned. When this happens, the consumer needs to react by requesting consumption of the same partitions that were assigned to it
- Once the ledger partitions have been consumed and the state is populated in memory, consumption of the stream topic can resume.
- There is no possibility of race conditions here as the instance that was responsible to populate the state in those particular ledger partitions had their stream partitions removed already as part of the second step, so consumption and processing have been stopped already

This diagram shows a potential initial state of an application exploiting this responsible to populate the state in those particular ledger partitions had their stream partitions removed already as part of the second step, so consumption and processing have been stopped already

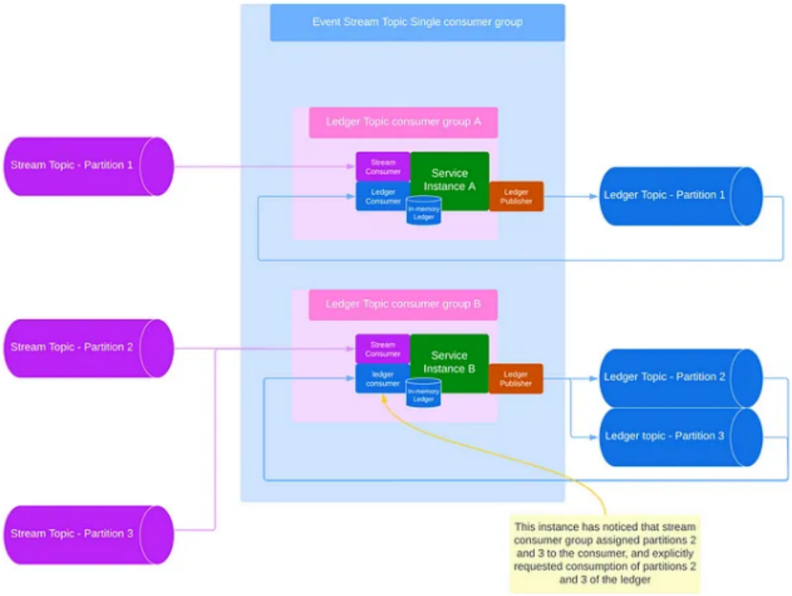
This diagram shows a potential initial state of an application exploiting this pattern:


Event Stream Topic Single consumer group
Ledger Topic consumer group A
Stream Topic - Partition 1
Stream Consumer
Ledger Consumer
Service Instance A
In-memory Ledger
Ledger Publisher
Ledger Topic - Partition 1
Ledger Topic consumer group B
Stream Topic - Partition 2
Stream Consumer
Ledger Consumer
Service Instance B
In-memory Ledger
Ledger Publisher
Ledger Topic - Partition 2
Ledger Topic consumer group C
Stream Topic - Partition 3
Stream Consumer
Ledger Consumer
Service Instance C
In-memory Ledger
Ledger Publisher
Ledger Topic - Partition 3

Publishers don't need any specific configuration as long as partition number and partitioning logic of the two topics is the same (and the same key is used)

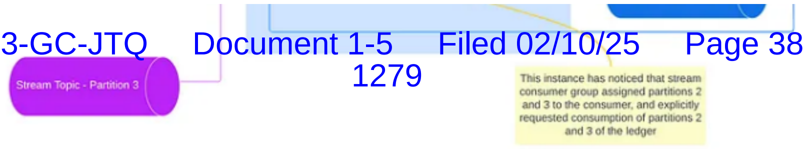
Design Diagram for state sharding in Kafka — initial state

While this one shows the state of the system after a rebalancing caused by the removal of one of the instances :


Event Stream Topic Single consumer group
Ledger Topic consumer group A
Stream Topic - Partition 1
Stream Consumer
Ledger Consumer
Service Instance A
In-memory Ledger
Ledger Publisher
Ledger Topic - Partition 1
Ledger Topic consumer group B
Stream Topic - Partition 2
Stream Consumer
ledger consumer
Service Instance B
In-memory Ledger
Ledger Publisher
Ledger Topic - Partition 2
Ledger topic - Partition 3
Stream Topic - Partition 3

This instance has noticed that stream consumer group assigned partitions 2 and 3 to the consumer, and explicitly requested consumption of partitions 2 and 3 of the ledger

Design Diagram for state sharding in Kafka — state after removal of a consumer


Stream Topic - Partition 3
This instance has noticed that stream consumer group assigned partitions 2 and 3 to the consumer, and explicitly requested consumption of partitions 2 and 3 of the ledger

Design Diagram for state sharding in Kafka — state after removal of a consumer

A deeper look on various concerns

Performance

- **In Kafka there is usually a tradeoff between throughput and latency.** For moderate traffic there is usually no need to delve deep into this, but when the amount of data to process becomes very high it is important to properly configure the system to behave exactly as expected

- **Producer Configuration —** Every producer can have a batch size defined that can be configured, and that would allow the client to 'batch' multiple messages into a single request to the server. The higher the batch size, the higher the potential throughput of the application, since there will be fewer trips to the server. However, a batch size too large could impact the memory consumption of the application and could even increase latency, since the client will linger for a certain period of time waiting for messages to accumulate. **properties :** batch.size — linger.ms — max.request.size

- **Consumer Configuration —** Consumers can define different configurations based on the use case. For example, it is possible to configure a minimum amount of data to be returned from the server to allow accumulation of more messages before returning to the client. This could increase throughput, at the cost of additional latency. **properties :** fetch.min.bytes — fetch.max.bytes — max.partition.fetch.bytes — fetch.max.wait.ms

- **Timeouts and heartbeats —** How long the broker should wait for confirmation before starting the rebalancing process, in the case of a consumer losing connection (or crashing). We want to be reactive in these cases to quickly recover proper state of processing, however we do not want to cause unnecessary rebalancing in case of a network blip that would be solved quickly. **properties :** session.timeout.ms — heartbeat.interval.ms

- **Compression —** Kafka allows messages to be compressed in several codecs, to reduce both the storage needed and the network bandwidth, at the cost of more CPU cycles and a slight delay in message dispatch. There are several codecs to choose from that all define different balances between gains and tradeoffs. Compression effectiveness also depends on producer batching configuration (the higher the batch size, the more effective compression will result, especially if the data is fairly repetitive, like XML or JSON data). **properties :** compression.type

> *there are deep dives available online, one example is this : Message compression in*

between gains and tradeoffs. Compression effectiveness also depends on producer batching configuration (the higher the batch size, the more effective compression will result, especially if the data is fairly repetitive, like XML or JSON data). **properties :** compression.type

> *there are deep dives available online, one example is this : Message compression in Apache Kafka — IBM Developer .*

There are several other considerations to make in regard to performance tuning that would deal with:

- **proper number of partitions based on the use case** — usually a higher number of partitions would ensure a higher potential throughput, but too many partitions may have significant tradeoffs in term or rebalancing speed or cluster performances.
- **partition replication and producer acknowledgment** — partition replica would ensure better availability in case of broker failures, but acknowledgment of confirmation for all replica can be costly in terms of producers' throughput.
- **producer and consumer buffer size** — a higher buffer size for clients would allow for a potential higher throughput and diminished latency, or even better resiliency in case of unexpected spikes of messages to be produced/consumed, but at cost of more resources necessary for the clients.
- **data retention policies, segment size, and cleanable dirty ratio** — smaller segment sizes and highest dirty ratio would ensure that topics are better maintained and unneeded data is removed quickly, but this could be costly for the broker to maintain since it increases the amount of work in the background.

> *A good starting point for some of these configurations can be this article : Deep dive into Apache Kafka storage internals: segments, rolling and retention (strimzi.io)*
>
> *All the configuration details for the broker, the topics, the producers and the consumers can be found here: Apache Kafka Configuration*

Fault tolerance

Given the rebalancing system that the broker has in place, with multiple services deployed, the solution is able to withstand failures of one or more consumers automatically. The result of rebalancing is a short interruption of processing of the partitions that are being rebalanced. There are different ways to manage these cases, and they depend on how we have configured the consumer to 'confirm' to the broker that a message has been processed. Kafka can be configured to confirm the consumption of a message to the broker explicitly for each message, or instead have a asynchronous processing of the partitions that are being rebalanced. There are different ways to manage these cases, and they depend on how we have configured the consumer to 'confirm' to the broker that a message has been processed. Kafka can be configured to confirm the consumption of a message to the broker explicitly for each message, or instead have a asynchronous automatic commit system that sends the confirmation in bulk at an interval. If the consumer does not have a chance of gracefully shut down, there is the possibility that some of the processed messages could be consumed again after rebalancing has completed. The consumers need to take this into account and define an AtLeastOnce consuming approach to be able to deal with potential duplicates correctly. **properties :** enable.auto.commit — auto.commit.interval.ms

As in every microservice architecture, the overall health of the system depends on the health of the individual services and being capable of handling failures is an integral part of every solution design. When dealing with Kafka applications, there are other things to take into consideration apart from the consumer health that we just discussed. A consumer service may need to confirm that the producer of the messages it expects is healthy. This is often done by implementing a 'heartbeat' kind of message in the stream, that the consumer expects at regular intervals even if there are no other messages generated as part of the normal flow. Failure to receive this heartbeat message may mean that the producer is unavailable and thus the system may not be healthy as a whole. Usually, a heartbeat message is expected every few seconds, and the consumer defines a grace period after which it needs to react based on the use cases, by sending an alert to a monitoring system, or by stopping processing any internal work that relies on data being correctly up to date. For example, a processing stream that relies on real-time information for a sport event may react to this by stopping accepting bets in case the heartbeat message is not received, since there is a reasonable suspicion that we are failing to get the latest information about the event.

> *There is a dedicated DK article that explains in detail how components manage resiliency in case of failures of external dependencies: Resiliency in Feeds Integration | DraftKings Engineering | Medium*

Even tough Kafka ensures data retention via partition replica, at DraftKings there is an additional layer of resiliency. Compacted topics that are expected to be the single source of truth are also backed up at regular intervals (usually once per day). This is done in order to ensure disaster recovery capabilities if for any reason the production cluster gets corrupted or is completely unavailable and data cannot be recovered easily. Not every topic has a backup. Most of the data does not need to be recovered this way as it can always be retrieved from other sources. This backup is more of an exception than a norm.

completely unavailable and data cannot be recovered easily. Not every topic has a backup. Most of the data does not need to be recovered this way as it can always be retrieved from other sources. This backup is more of an exception than a norm.

Monitoring and alerting

Kafka exposes a good number of metrics to be monitored to ensure the system is healthy. It exposes commit latency, used buffer size, rebalancing latency, throughput of producers, and much more. A complete list of metrics is available at Apache Kafka / Monitoring . The client libraries expose a way to retrieve these metrics to be sent to any monitoring solution. At DraftKings, App Metrics backed up by Grafana or Datadog is the most commonly used framework.

However, one of the main metrics used is simply the latency of processing. Each Kafka message has a timestamp that defines the moment in which the producer has produced the message. By comparing this timestamp against the one recorded when the consumer has received the message, and then again once the message has been processed, we can have a clear number explaining the consumer delay and the processing time for each message. With this minimal setup, is pretty simple to understand if the consumer is 'falling behind' and is not able to catch up with the rate of messages being produced upstream.

Another valid way of measuring consumption delay is comparing the latest committed offset of a consumer group on each partition against the latest produced offset: also in this case, if the gap between the two values increases too much, an automatic alert can be implemented in order to notify the possibility of data being delayed over an accepted threshold.

These alerts could be then used to evaluate how to improve the system — for example, scaling up or scaling down the consumer, change the configuration at cluster, topic, producer or consumer level, increase or decrease the resources dedicated to each instance of the service, and so on.

Want to learn more about DraftKings' global Engineering team and culture? Check out our Engineer Spotlights and current openings!

Kafka State Management Software Architecture

13

Published in DraftKings Engineering

Follow

285 Followers · Last published Oct 2, 2024

DraftKings Engineering

Published in DraftKings Engineering

Follow

285 Followers · Last published Oct 2, 2024

DraftKings Engineering

Written by Nicola Atorino

Follow

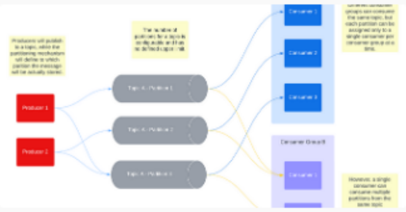
21 Followers · 7 Following

No responses yet

What are your thoughts?

Respond

More from Nicola Atorino and DraftKings Engineering

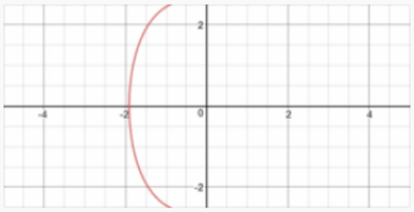


In DraftKings Engineering by Nicola Atorino

Apache Kafka as a centralized state management platform — Pa...

Introduction

Jul 7, 2023 13 1

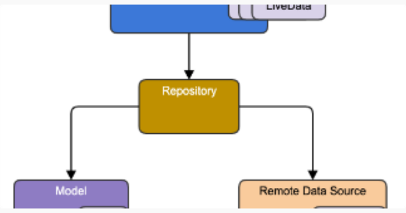


In DraftKings Engineering by Mehrad Kavian

Signature Malleability

Many smart contracts rely on the validity of a signature or whether it has been used befor...

Jun 20, 2023 30



In DraftKings Engineering by Mike DeMaso

Android, MVVM, and Repositories in the Real World

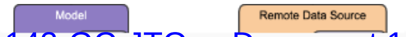
It is a tale as old as Android Development itself. How do we achieve a responsive user

liveness healthcheck endpoint
readiness healthcheck endpoint
get last status
get last status
business logic start / stop
Readiness HealthStatus Changed Event

In DraftKings Engineering by Nicola Atorino

Resiliency in Feeds Integration

Learn how the Feeds Integration team in DraftKings handles unexpected situations...



In DraftKings Engineering by Mike DeMaso

Android, MVVM, and Repositories in the Real World

It is a tale as old as Android Development itself. How do we achieve a responsive user…

Nov 2, 2020 21

In DraftKings Engineering by Nicola Atorino

Resiliency in Feeds Integration

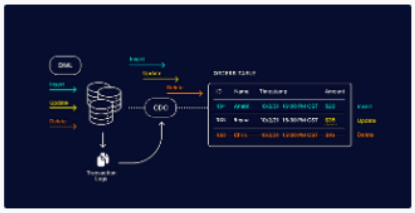
Learn how the Feeds Integration team in DraftKings handles unexpected situations…

Jan 6, 2023 61

See all from Nicola Atorino

See all from DraftKings Engineering

Recommended from Medium



In Level Up Coding by Jacob Bennett

The 5 paid subscriptions I actually use in 2025 as a Staff Software…

Tools I use that are cheaper than Netflix

Jan 7 6.9K 146

Luis Soares

The Change Data Capture (CDC) Design Pattern

Change Data Capture (CDC) is a design pattern that identifies and tracks changes in…

Jan 16, 2024 797 6

Lists

Stories to Help You Grow as a Software Developer

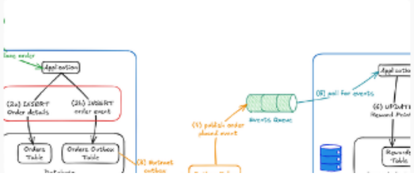
19 stories · 1572 saves

General Coding Knowledge

20 stories · 1886 saves

Staff picks

805 stories · 1590 saves



Jan 7 6.9K 146

Jan 16, 2024 797 6

Lists

Stories to Help You Grow as a Software Developer

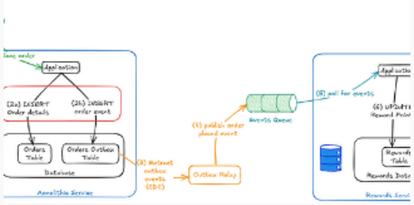
19 stories · 1572 saves

General Coding Knowledge

20 stories · 1886 saves

Staff picks

805 stories · 1590 saves

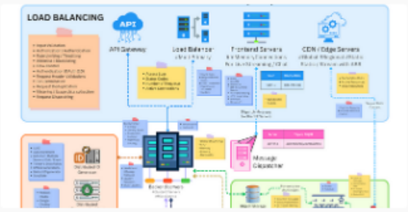


In Dev Genius by Laksh Chauhan

Distributed Transaction Patterns in Event-Driven Microservices

Design patterns to maintain consistency for an event spanning across services in a...

Jan 5 191 3



In ByteByteGo System Design Alli... by Love Shar...

System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However,...

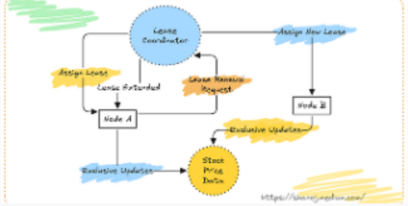
Sep 17, 2023 9K 57

In Coding Beauty by Tari Ibaba

This new IDE just destroyed VS Code and Copilot without even...

Wow I never thought the day I stop using VS Code would come so soon...

Jan 17 1.8K 73



In Stackademic by Shanoj

Distributed Systems Design Pattern: Lease-Based Coordinati...

The Lease-Based Coordination pattern offers an efficient mechanism to assign temporary...

Dec 17, 2024 28

See more recommendations

Help Status About Careers Press Blog Privacy Terms Text to speech Teams

Exhibit 64

Apache Kafka as a centralized state management platform — Part 1

Nicola Atorino · Follow
Published in DraftKings Engineering · 8 min read · Jul 7, 2023

13 1

Introduction

Overview & Capabilities of Apache Kafka

The main mission of the Sportsbook domain in DraftKings (DK) is to provide a comprehensive sport offering, near real-time, to allow customers to enjoy their experience with a constant growing set of products, events, and capabilities. To handle this, the underlying infrastructure that powers the entire sportsbook system must adhere to strict criteria in terms of performance, stability, and fault-tolerance. Most of DK's infrastructure is designed to be horizontally scalable, in order to accommodate for high volumes and to leave room for further growth, when necessary.

In order to reach the desired state, the DK architecture team needed a way to manage the data flow that would allow for high flexibility, low latency, and high stability. While the entire infrastructure uses several different technologies in different contexts, the focus of this article will be on one of the most used tools, Kafka.

Apache Kafka is an open source, distributed streaming platform that is used for building real-time data pipelines and streaming applications. It is designed for high scalability and high throughput, and can handle millions of events per second. Kafka is used for various use cases such as logging, data streaming, event sourcing, and more. It also provides durable, fault-tolerant storage and can be integrated with other systems and tools.

Kafka leverages a wide variety of patterns. The first one that usually comes into mind is data streaming, for communication between two different services, and that is one of the patterns that is widely used at DK. However, Kafka also has high potential as a platform for caching and state management. We're going to explore 3 different patterns:

- distributor of data stored in a SQL database for read-only purposes
- centralized persistent layer for distributed applications

management. We're going to explore 3 different patterns:

- distributor of data stored in a SQL database for read-only purposes
- centralized persistent layer for distributed applications
- data sharding in order to allow for distributed processing

Before diving into these use cases, let's have a small recap of some of the concepts leveraged by the architecture team when designing systems around Kafka.

Partitioning and Consumer Groups

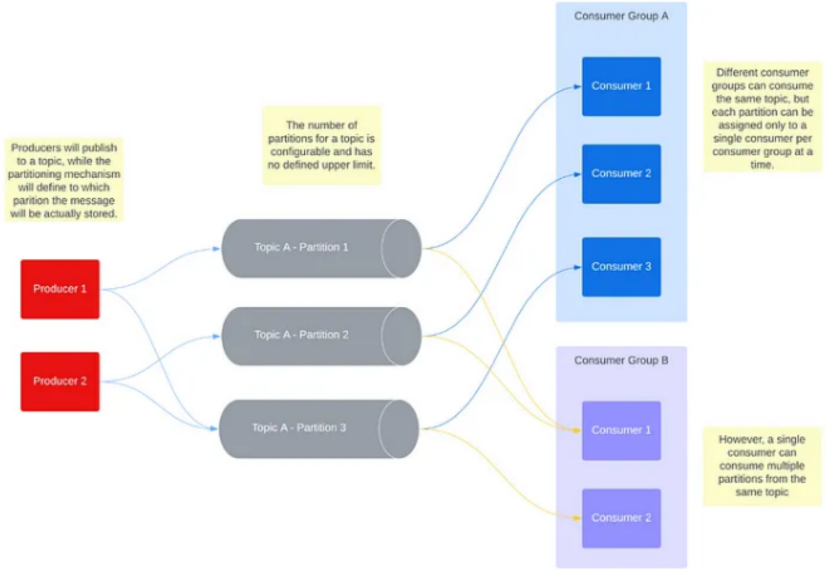
Kafka works by allowing producers to publish a stream of messages in a topic, that is a logical grouping of different files (called 'partitions') where the actual messages are stored. It is essentially an append-only log where each message contains a key, a value, and optionally some headers. When a publisher pushes a message to a topic, the actual partition to which the message is pushed is defined by a partitioning mechanism usually based on the message key, although it may be possible to implement different partitioning logics. Typically, in our domain, partitioning is done based on the key, so messages with the same key will be stored in the same partition.

The concept of partitioning is important when it comes to consumption distribution: a set of consumers can leverage partitioning to evenly distribute the load. When a consumer registers to a topic, it provides the Kafka broker with a key called 'consumer group' that will identify the group this specific consumer is part of. Multiple consumers sharing the same group will distribute the consumption of the topic, sharing the partitions between them. This way, a topic that contains multiple partitions can be split between multiple consumers, with the upper limit being the total number of partitions the topic is composed of. Since a partition can only be consumed by a single consumer at a time for each consumer group, this guarantees ordering in the consumption of the messaging while allowing horizontal scalability.

Each consumer group can optionally register an offset, or "pointer," in the partition, indicating the last message that was correctly consumed by any consumer of the consumer group. When a partition switches consumers within the group (as a result of changes in the topology of the system, redeployment, or scaling) the new instance will resume consumption starting from the last committed offset left from the previous consumer in the consumer group. This is made possible by a special node in the cluster that takes care of coordinating the consumer groups. If the offset is missing, the consumer can be configured to read the topic from the beginning or from the end, based on the use case that makes sense for that application.

More information can be found in the official Kafka documentation : https://kafka.apache.org/documentation/#impl_offsettracking that takes care of coordinating the consumer groups. If the offset is missing, the consumer can be configured to read the topic from the beginning or from the end, based on the use case that makes sense for that application.

More information can be found in the official Kafka documentation : https://kafka.apache.org/documentation/#impl_offsettracking

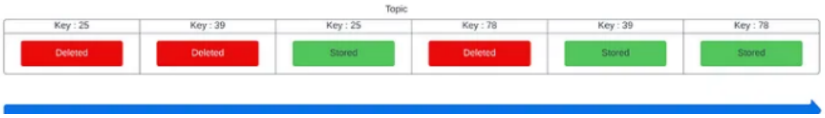

Producers will publish to a topic, while the partitioning mechanism will define to which parition the message will be actually stored.
The number of partitions for a topic is configurable and has no defined upper limit.
Consumer Group A
Different consumer groups can consume the same topic, but each partition can be assigned only to a single consumer per consumer group at a time.
Producer 1
Producer 2
Topic A - Partition 1
Topic A - Partition 2
Topic A - Partition 3
Consumer 1
Consumer 2
Consumer 3
Consumer Group B
Consumer 1
Consumer 2
However, a single consumer can consume multiple partitions from the same topic

Partition Balancing in Kafka

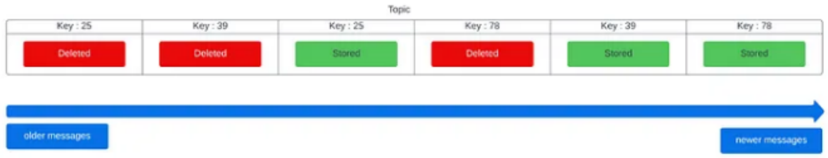
Retention Policies

Storage space is not infinite. One of the key features of Kafka is its ability to retain data for a specified period of time. This is known as the retention policy. Retention policies are used to control how long data is kept in a topic before it is deleted. It can be set on a per-topic basis, and defined using either time or size-based criteria. This way, our topics used for streaming data will keep messages only for a certain period of time. This is known as the 'delete' retention policy.
The most interesting policy for our use cases, though, is the 'compact' policy. When setting this policy on a topic, the cluster will delete duplicate messages, while keeping the latest or most recent version of each message. It is also incredibly useful to handle state management of an application, because there are several cases where we are interested only in the most recent version of the data, like when implementing a caching mechanism.

The two policies can be combined. A message can have all its duplicates removed and the most recent version can then be deleted from the topic following the delete retention policy.



following the delete retention policy.



Compact retention policy in Kafka

Although there are multiple other configurations and concepts to explore that would take a lot of time to cover properly, and several considerations to make when choosing the proper strategy in terms of partitioning and data retention, these are some of the concerns to have when planning an architecture using this technology:

- cluster performances
- storage
- producers' speed
- serialization
- consumption strategies

More information about log compaction can be found in the official Kafka documentation:
https://kafka.apache.org/documentation/#design_compactionbasics

Use of Kafka as a DB Cache Distributor

Several services in DK need to constantly evaluate incoming messages against the most recent version of data stored in the database. Given the number of services that require this info, a direct connection to the database for each one of these components would put too much strain on the database instance. Also, the amount of data that need to be retrieved by the application would generate too much internal traffic. To solve this situation, Kafka is used as a push-based mechanism to allow applications to store the data they need in memory, be notified when data changes, and be able to handle all their processing internally.

The distributor

A Cache Distributor is a microservice whose only responsibility is to interrogate the database at regular intervals using predefined queries. When the application starts, the entire dataset is retrieved and pushed into a dedicated Kafka topic. After initial load, only updates are retrieved and published, using the datetime of the last request that was successfully processed. The queries could involve complex joins of different tables, thus effectively generating a view that will be then exposed to every consumer, which won't need to connect directly into the database (or handle their own dedicated Kafka topic. After initial load, only updates are retrieved and published, using the datetime of the last request that was successfully processed. The queries could involve complex joins of different tables, thus effectively generating a view that will be then exposed to every consumer, which won't need to connect directly into the database (or handle their own caching system) as long as the staleness of the data is acceptable for their purpose. On less frequent intervals, the entire dataset is fetched and republished in order for the older data to not be removed from the topic due to retention. The interval for re-fetching the entire dataset must be shorter than the topic's retention period to ensure that consumers do not miss infrequently updated data.

Topic configuration

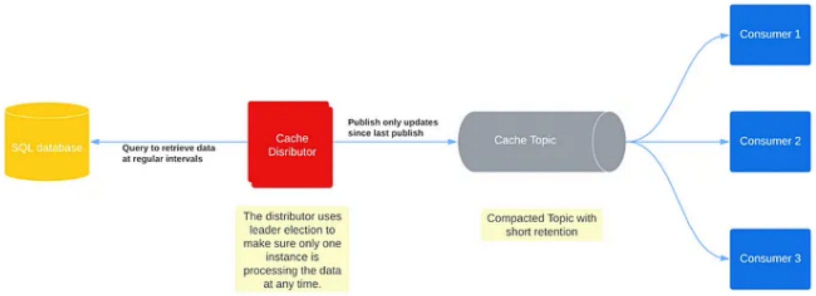
The data is pushed to a compacted topic, with the key being the primary key of the main entity that was retrieved. This way, when compaction is executed, only the most recent version of the item will be kept in the topic. When a consumer needs to read the entire topic in order to populate its own internal state on startup, they will avoid consuming a huge number of messages while ignoring everything apart the last one, speeding up the startup time.

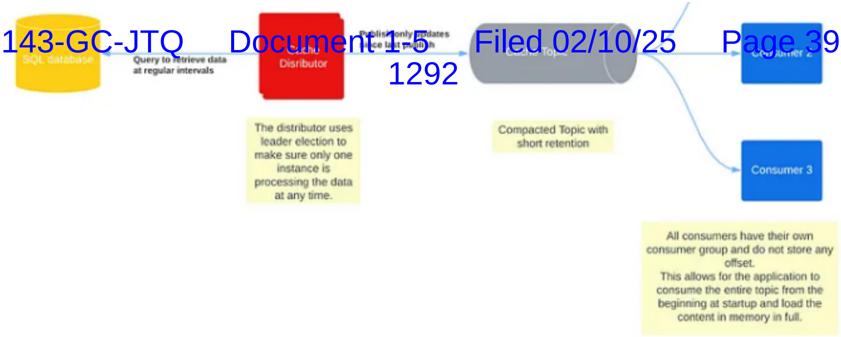
The topic also contains a delete policy retention, that will remove data after a certain amount of time. This is necessary to clean the topic from entities that have been deleted from the database.

The cache topic has a single partition. This is done because, in this particular use case, every consumer should consume the topic in full to maintain the entire cache in memory. By having a single partition, the broker performances improve. since it has to deal with less communication overhead.

The consumer

On the consumer side, the data is stored in memory. At startup, the topic is read from the beginning, and the messages are used to create an internal memory dictionary. If necessary, each application can create additional custom indexes to query the data based on their own needs. After the initial startup, the consumer keeps listening to the topic to receive regular updates, as they are going to be published by the producer.


SQL database
Query to retrieve data at regular intervals
Cache Disributor
Publish only updates since last publish
Cache Topic
Consumer 1
Consumer 2
Consumer 3
The distributor uses leader election to make sure only one instance is processing the data at any time.
Compacted Topic with short retention


SQL database
Query to retrieve data at regular intervals
Cache Disributor
Publish only updates since last publish
Cache Topic
Consumer 2
Consumer 3

The distributor uses leader election to make sure only one instance is processing the data at any time.

Compacted Topic with short retention

All consumers have their own consumer group and do not store any offset.
This allows for the application to consume the entire topic from the beginning at startup and load the content in memory in full.

Design Pattern of the DB Cache Distributors

Considerations

This system has several advantages, but also some drawbacks that need to be pointed out. First of all, it is memory intensive for the consumers, but this is true for every state management system that involves the use of Kafka, including the ones being discussed next. The full reload that happens on regular intervals can cause memory spikes while the application processes the set and discards the old values for the updated ones. This can be mitigated by reducing the amount of data the service stores from the topic to the minimum. If the cache contains an object with a lot of properties, but not all of them are necessary, it is better to define a lightweight object on the consumer side and store only what is needed.

Secondly, the staleness of the data cannot be configured at the consumer side, only at the distributor side. If a consumer requires data that is more up to date than what the cache provides, it will have to handle this in different ways. One of the classical problems found with this approach is the retrieval of information from the database that was changed as a result of the processing of the microservice itself. A possible workaround for this would be to update the internal dictionary (if all the data is known) before submitting the database change, but this can lead to data inconsistencies (especially if the update needs to be validated downstream) and is generally regarded as a sub-optimal solution.

In short, this system is not suitable to be used with components that are expected to update into the database (directly or indirectly) the same set of data they are going to retrieve from the cache.

Next

In the next article, we're going to explore two other patterns for state management that will overcome some of the downsides that we encountered till now:

- Kafka as a centralized store for distributed applications
- Kafka used to handle state sharding

The second part can be found here : https://medium.com/draftkings-

- Kafka as a centralized store for distributed applications
- Kafka used to handle state sharding

The second part can be found here : https://medium.com/draftkings-engineering/apache-kafka-as-a-centralized-state-management-platform-part-2-980631412a5a

Want to learn more about DraftKings' global Engineering team and culture? Check out our Engineer Spotlights and current openings!

Kafka | State Management | Software Architecture

13 | 1

Published in DraftKings Engineering

Follow

285 Followers · Last published Oct 2, 2024

DraftKings Engineering

Written by Nicola Atorino

Follow

21 Followers · 7 Following

Responses (1)

What are your thoughts?

Respond

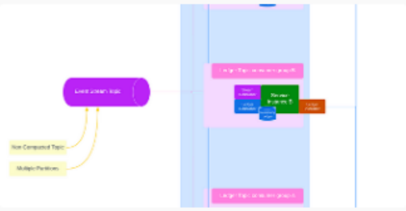
D Dima Paikin
Jul 18, 2023

Why not use Kafka Connect instead of a Cache Distribution service?

1 reply Reply

More from Nicola Atorino and DraftKings Engineering

More from Nicola Atorino and DraftKings Engineering

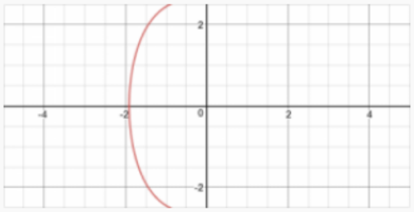


In DraftKings Engineering by Nicola Atorino

Apache Kafka as a centralized state management platform — Pa…

Kafka for centralized storage and state sharding

Jul 7, 2023 👏 13

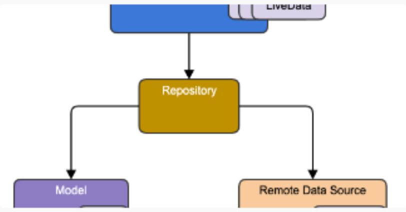


In DraftKings Engineering by Mehrad Kavian

Signature Malleability

Many smart contracts rely on the validity of a signature or whether it has been used befor…

Jun 20, 2023 👏 30



In DraftKings Engineering by Mike DeMaso

Android, MVVM, and Repositories in the Real World

It is a tale as old as Android Development itself. How do we achieve a responsive user…

Nov 2, 2020 👏 21

In DraftKings Engineering by Nicola Atorino

Resiliency in Feeds Integration

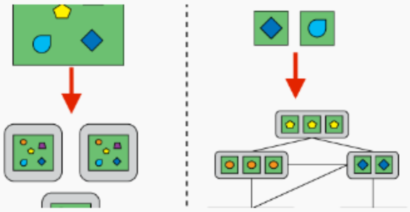
Learn how the Feeds Integration team in DraftKings handles unexpected situations…

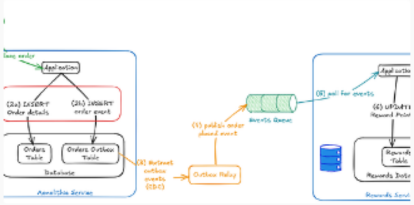
Jan 6, 2023 👏 61

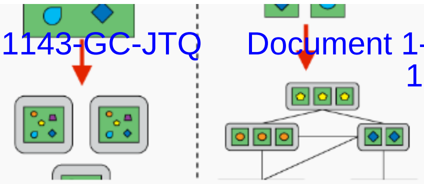
See all from Nicola Atorino

See all from DraftKings Engineering

Recommended from Medium





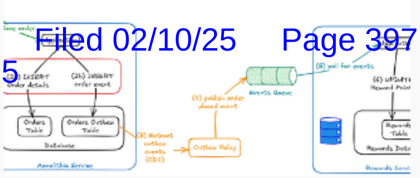


In Artificial Intelligence in Plain Eng... by Amit Sin...

Service-Oriented Architecture (SOA) vs. Microservices

In today's fast-evolving tech landscape, building scalable, maintainable, and flexible...

Sep 26, 2024 · 175 · 1



In Dev Genius by Laksh Chauhan

Distributed Transaction Patterns in Event-Driven Microservices

Design patterns to maintain consistency for an event spanning across services in a...

Jan 5 · 191 · 3

Lists

Stories to Help You Grow as a Software Developer

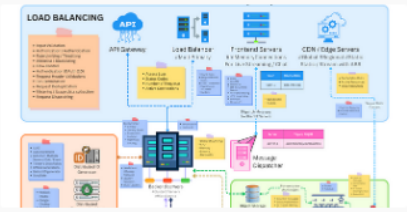
19 stories · 1572 saves

General Coding Knowledge

20 stories · 1886 saves

Staff picks

805 stories · 1590 saves

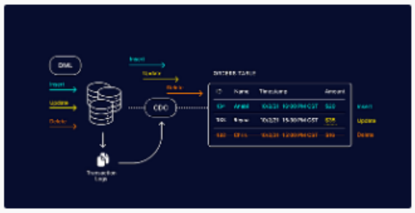


In ByteByteGo System Design Alli... by Love Shar...

System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However,...

Sep 17, 2023 · 9K · 57



Luis Soares

The Change Data Capture (CDC) Design Pattern

Change Data Capture (CDC) is a design pattern that identifies and tracks changes in...

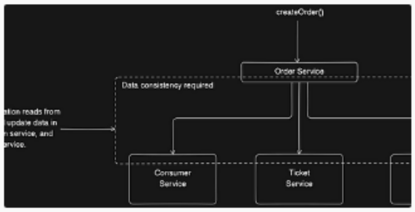
Jan 16, 2024 · 797 · 6

Ali Zeynalli

How to Become a Strong Software Architect

Roadmap of a Software Architect

Jan 3 · 611 · 13



Joud W. Awad

Microservices Pattern: Distributed Transactions (SAGA)

Explore the SAGA Pattern: Ensuring Data Integrity in Microservices. Dive into its...

Jun 16, 2024 · 1.3K · 9

✦ Sep 26, 2024 👏 175 ● 1 ✦ Jan 5 👏 191 ● 3

Lists

Stories to Help You Grow as a Software Developer

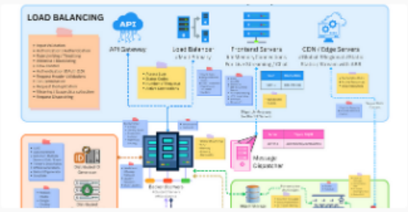
19 stories · 1572 saves

General Coding Knowledge

20 stories · 1886 saves

Staff picks

805 stories · 1590 saves

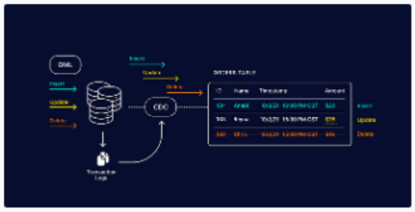


In ByteByteGo System Design Alli... by Love Shar...

System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However,...

✦ Sep 17, 2023 👏 9K ● 57



Luis Soares

The Change Data Capture (CDC) Design Pattern

Change Data Capture (CDC) is a design pattern that identifies and tracks changes in...

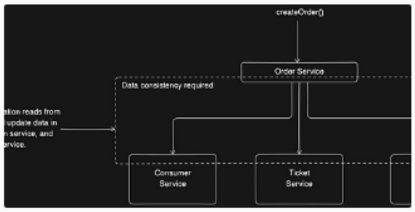
✦ Jan 16, 2024 👏 797 ● 6

Ali Zeynalli

How to Become a Strong Software Architect

Roadmap of a Software Architect

✦ Jan 3 👏 611 ● 13



Joud W. Awad

Microservices Pattern: Distributed Transactions (SAGA)

Explore the SAGA Pattern: Ensuring Data Integrity in Microservices. Dive into its...

✦ Jun 16, 2024 👏 1.3K ● 9

See more recommendations

Help Status About Careers Press Blog Privacy Terms Text to speech Teams

Exhibit 65

GGB NEWS EXECUTIVE SUMMARY

Vol. 23 • No. 03 • January 31, 2025 Print the Current Issue

April 8, 2024 , SPORTS BETTING, Featured Stories

Wizards of Odds: The Science and Hard Work of Trading Teams

By Cole Rush Mon, Apr 8, 2024 SHARE ON: Twitter Facebook LinkedIn

In order for sportsbooks to function properly, they require dedicated teams of traders to constantly tinker with, analyze and produce the best possible odds. It's a tough job, one that requires critical thinking, a knack for numbers and most of all, a love for the game.

The sports betting whirlwind spirals inevitably to new and established markets alike, bringing with it a storm of wagering activity. In the eye of the cyclone sit trading teams—dedicated groups of professionals pulling the strings to keep sportsbook operations afloat amidst the high winds of tentpole events and niche competitions alike.

Behind the Curtain

Pull back the veil on a trading team, and you're likely to see a metaphorical array of bells and whistles, buttons and levers, all being pulled, pushed and triggered at various intervals. It's a complex orchestra with numerous conductors waving their batons in rhythmic synchronization.

Any given sports bettor won't notice much of a trader's work, and that's by design. Traders are the wizards working their magic—which, in fact, is just technology and applied knowledge—to make betting possible.

Georgios Ioakeimidis, senior trading operations manager at OpenBet, says: "We have traders with various focuses. Some handle our Managed Trading Services (MTS) customers, while others perform in-play duties or risk management. There isn't one specific role; rather, there are different aspects across different departments."

A typical day for a trader, Ioakeimidis continues, involves "checking prices, events, and markets. We closely monitor information that may affect these aspects and adjust accordingly. Utilizing tools to monitor risk is crucial as each event is unique, with every day presenting new challenges."

Chase Thomson, account executive at Simplebet, says minor adjustments and quick tweaks keep most of the secret sauce of trading invisible to bettors, and that's a good thing.

"In an ideal world, [trading] is very under the radar," he says. "There will be the occasional price adjustment, but we don't have to adjust too frequently because our modeling is very accurate, and ultimately, we want to allow as much time as possible for the players to place bets without getting rejected."

Simplebet traders begin their day by looking over the results of the previous day, logging any discrepancies and noting observations to apply in the future.

"Following this," Thomson says, "the team goes straight into researching the upcoming fixtures for that day or evening. For Simplebet, that means looking at team tendencies, as

What should regulators and operators do to control the growth of sweepstakes social casinos?

○ Shut them all down state by state

○ Identify the bad apples and shut them down via regulatory challenges

○ Establish regulations, tax rates, AML, KYC and RG programs that will guarantee integrity

○ Threaten to boycott the payment providers who work with sweepstakes casinos

GAMING EVENTS CALENDAR

January 27-29: AML Fundamentals + AML for Casinos, Hilton San Diego Airport/Harbor Island. Gain insights into latest AML practices, regulations, and challenges for casinos. Produced by Skilld. For more information visit skilld.com/aml-event-2025 or contact events@skilld.com/888.333.3039.

bets without getting rejected.

Simplebet traders begin their day by looking over the results of the previous day, logging any discrepancies and noting observations to apply in the future.

"Following this," Thomson says, "the team goes straight into researching the upcoming fixtures for that day or evening. For Simplebet, that means looking at team tendencies, as well as bettor trends for the teams that are playing that day. This includes looking into if bettors bet on positive outcomes or negative outcomes for those teams and reviewing team's general trends and approaches, such as 4th down aggression, run vs pass tendencies, and more."

While games are in progress, traders then monitor the action and check all the data for accuracy throughout the day. They can also move or suspend lines if needed.

All of this behind-the-scenes work translates to a seamless and simple experience for the user. Malte Hegelr, VP of product at BtoBet, says: "Bettors, through our customers, have access to over 75 sports selections, including a wide range of niche sports and 2,000 unique betting markets. Our work ensures a wide spectrum of choices."

No Place Like Home

Trading is a high-octane and high-pressure job. It's no surprise, then, that most traders are passionate about their work.

"Traders possess an in-depth knowledge of the leagues, teams, players, and market dynamics," says OpenBet's Ioakeimidis. "They can make informed decisions about adjusting odds."

In other words, traders are fans. They like sports, and they have a deep wealth of expertise to draw from. Market analysis tools and monitoring software help fill the gaps.

The passion and knowledge required is a good foundation, but traders need more to succeed. Their job blends heart, brains and courage to make quick decisions. Communication is key.

"Our team of traders maintains constant communication with each other through various channels," says BtoBet's Hegelr. They also communicate with our clients through various channels to maintain transparency and address their needs promptly."

Careful planning can also prepare traders for chaotic days or particular events. Thomson of Simplebet says: "Most communication actually happens before the event starts. Traders are well-trained and are directed on what to do in every situation before the games commence. Once things start moving, everyone on the trading floor is in constant communication."

Not in Kansas Anymore

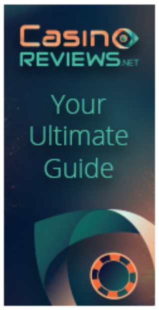
With the building blocks of a solid trading team in place, the challenge then becomes bringing trading services into legal markets. It's hard enough to manage one market, let alone a smattering of global markets with distinct regulations and policies.

"The main focus around regulation is to not offer a market with the outcome already determined," says Thomson. "We offer over 1,000 markets in most of our games, and making sure that none of those markets are up with a determined outcome is extremely important."

Geography obviously comes into play for traders, such as in the U.S. market, where college sports are both highly popular and highly regulated. College sports regulations vary widely. As of writing, Ohio prohibits player props on NCAA sports, Illinois bars any bets on in-state schools and other states have their own minutiae for traders to parse.

"In the U.S., this is further compounded by ensuring that no markets are offered on certain college teams and jurisdictions, so we need to be extremely vigilant with what we do," says Thomson."

Regulations also encompass external factors. Ioakeimidis says: "Age limits, permissible sports and markets, responsible gambling practices, and consumer protection" are all top considerations for a trading team.



FEATURED STORIES

CEDAR RAPIDS CASINO HOPES DWINDLING AS MORATORIUM ADVANCES

WEEKLY FEATURE: OPTIMISM FOR BRAZIL CASINO LEGALIZATION IN H1 2025

U.S. IN FOCUS

SPORTS BETTING IN FOCUS

TRIBAL GAMING IN FOCUS

ASIA IN FOCUS

LATIN AMERICA IN FOCUS

SUPPLIERS IN FOCUS

FANTINI'S FINANCE: BACK AGAINST THE WALL

GGB PODCAST: DEREK WEBB, TABLE GAME INVENTOR; FOUNDER, THE CAMPAIGN FOR FAIRER GAMBLING

TABCORP CONFIRMS CEO ARRIVAL

ATLANTIC LOTTERY APPOINTS MCCREADY AS NEW CEO

GROUPE PARTOUCHE OPENS BENIN CASINO

“The main focus around regulation is to not offer a market with the outcome already determined,” says Thomson. “We offer over 1,000 markets in most of our games, and making sure that none of those markets are up with a determined outcome is extremely important.”

Geography obviously comes into play for traders, such as in the U.S. market, where college sports are both highly popular and highly regulated. College sports regulations vary widely. As of writing, Ohio prohibits player props on NCAA sports, Illinois bars any bets on in-state schools and other states have their own minutiae for traders to parse.

“In the U.S., this is further compounded by ensuring that no markets are offered on certain college teams and jurisdictions, so we need to be extremely vigilant with what we do,” says Thomson.”

Regulations also encompass external factors. Ioakeimidis says: “Age limits, permissible sports and markets, responsible gambling practices, and consumer protection” are all top considerations for a trading team.

Collaboration, too, is a defining factor in trading success. “We work closely with our legal and compliance teams to ensure full adherence to regulatory requirements while optimizing our offerings to meet customer demand,” says Hegelr.

The Yellow Brick Road

Communication? Check. Passion and expertise? Check. Regulations and compliance? Check. What’s next?

This is where the trading journey begins. Games start, whistles are blown, and step-by-step (or brick-by-brick) trading kicks off in earnest—it really is step-by-step, as the experts interviewed noted that movements in odds are usually gradual.

“Common types of odds movements occur regularly in response to global betting markets, with gradual adjustments being the norm,” says Hegelr.

Challenges can arise along the way, but they’re horses of a different color.

Hegelr continues: “Drastic movements are rare but can occur due to factors such as injury reports, unexpected weather conditions, or significant changes in team dynamics, such as a new coach.”

Gradual movements can be handled by computers, provided the right data is input ahead of time. But traders still must maintain a steady hand. Simplebet’s microbetting odds markets are particularly dynamic, requiring a combination of man and machine to manage.

“Most odds that are changed directly by the trader are relatively small,” Thomson says. “Odds changes are determined by action on the market and research done before the game. If unbalanced action is coming in on a market, then odds are adjusted towards that action.”

Ioakeimidis says: “Odds movements are an inherent aspect of the trading process. While significant fluctuations in odds are less common in prominent leagues like the Premier League, they can be more prevalent in lower-profile events or niche markets.

“Ultimately, the acceptability of odds changes is determined by a combination of factors, including market demand, risk exposure, and maintaining competitive pricing. Striking the right balance between responsive pricing adjustments and preserving profit margins is a delicate yet essential aspect of effective sportsbook trading.”

Did you enjoy this article? **Opt in for all the latest from *GGB News!***

SPORTS BETTING IN FOCUS
TRIBAL GAMING IN FOCUS
ASIA IN FOCUS
LATIN AMERICA IN FOCUS
SUPPLIERS IN FOCUS
FANTINI’S FINANCE: BACK AGAINST THE WALL
GGB PODCAST: DEREK WEBB, TABLE GAME INVENTOR; FOUNDER, THE CAMPAIGN FOR FAIRER GAMBLING
TABCORP CONFIRMS CEO ARRIVAL
ATLANTIC LOTTERY APPOINTS MCCREADY AS NEW CEO
GROUPE PARTOUCHE OPENS BENIN CASINO
TROUBLED STAR ENTERTAINMENT AGREES A$60M DEAL TO SELL STAR SYDNEY EVENTS CENTRE
DOUBTS REMAIN OVER CASINO SPACE ALLOCATION IN THAI BILL

COLUMNS
ASIAN GAMING
CASINO-OLOGY
EUROPEAN GAMING
GLOBAL GAMING
GOODS & SERVICES
LATIN AMERICA
NUTSHELL
ONLINE GAMING
PEOPLE
PRESS RELEASES
SPORTS BETTING
TRIBAL GAMING
UNITED STATES GAMING

GGB PUBLICATIONS
GGB Magazine
GGB News
Casino Style
Tribal Government Gaming

GGB Magazine | GGB News | Casino Style | Tribal Government Gaming

Exhibit 66

Contest Type: Beginner - Overview (US)

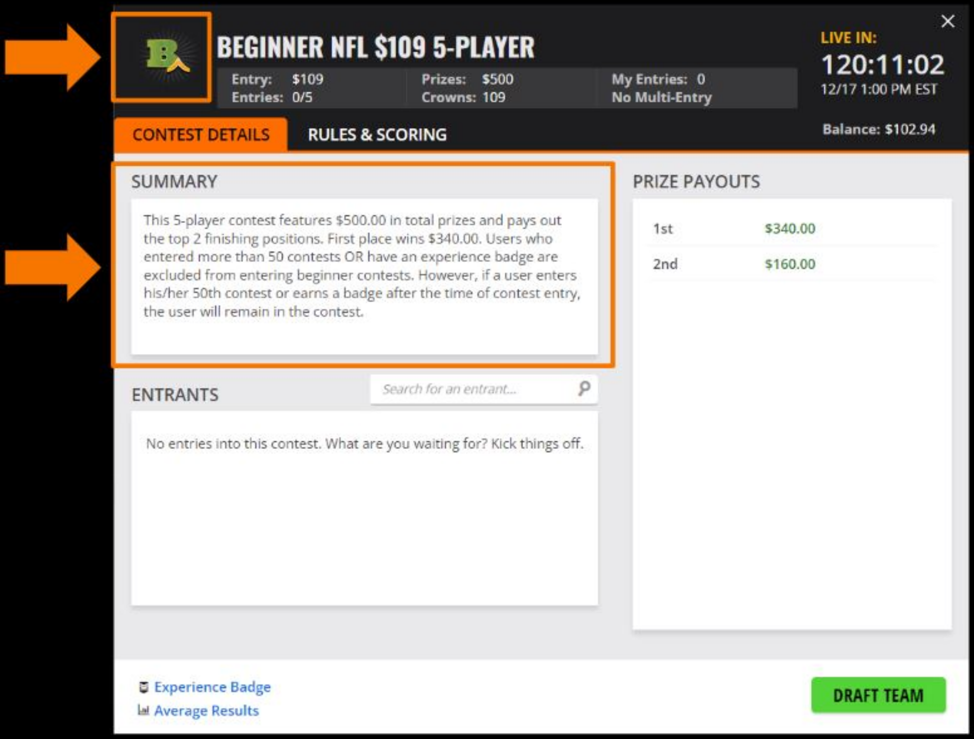
JUMP TO SECTION
Fantasy Sports Basics
How to Play- Fantasy Sports
Game Styles & Contest Types
Contest Rules & Scoring
Resolving Fantasy Sports Issues
Tickets
Leagues

What is a Beginner Contest Type?

Beginner Contest Types are available to players who have not yet earned an experience badge and have played under 50 contests.

- Learn more about experience badges.
- Learn more about who is eligible for Beginner or Casual Contests.

View this visual example below.



BEGINNER NFL $109 5-PLAYER

Entry: $109 Entries: 0/5 Prizes: $500 Crowns: 109 My Entries: 0 No Multi-Entry

LIVE IN: 120:11:02 12/17 1:00 PM EST

Balance: $102.94

SUMMARY

This 5-player contest features $500.00 in total prizes and pays out the top 2 finishing positions. First place wins $340.00. Users who entered more than 50 contests OR have an experience badge are excluded from entering beginner contests. However, if a user enters his/her 50th contest or earns a badge after the time of contest entry, the user will remain in the contest.

PRIZE PAYOUTS

1st	$340.00
2nd	$160.00

ENTRANTS

No entries into this contest. What are you waiting for? Kick things off.

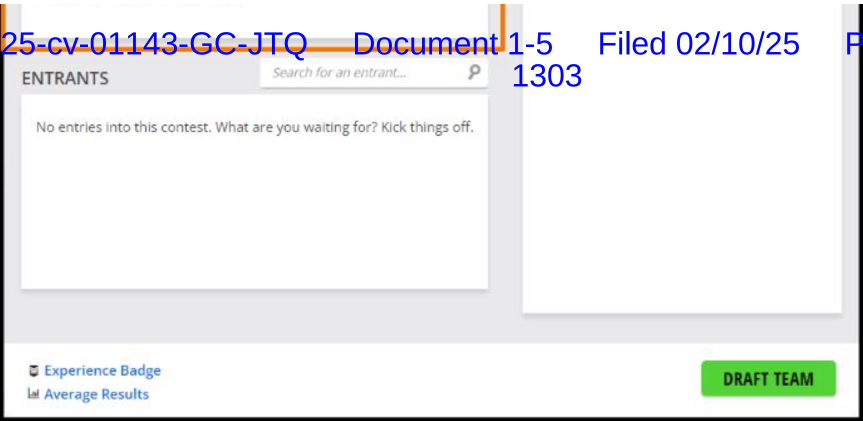
Frequently asked questions

Can I enter a Beginner Contest if I have an experience badge?

No, players with experience badges or who have entered more than 50 contests are not eligible to participate in Beginner Contests.

How can I find Beginner Contests?

How can we help?



ENTRANTS

Search for an entrant...

No entries into this contest. What are you waiting for? Kick things off.

Frequently asked questions

Can I enter a Beginner Contest if I have an experience badge?

No, players with experience badges or who have entered more than 50 contests are not eligible to participate in Beginner Contests.

How can I find Beginner Contests?

1. Filter or select your preferred **sport**.
2. Select the **Beginner** filter.
   - All available game styles will be displayed, including Classic and Showdown contests (where available).

What if I earn an experience badge while I am in a beginner contest?

You would remain in the Beginner Contest until the end of the contest, however, you wouldn't be able to enter another Beginner Contest after receiving an experience badge.

Was this article helpful? Yes No

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

How can we help?

Exhibit 67

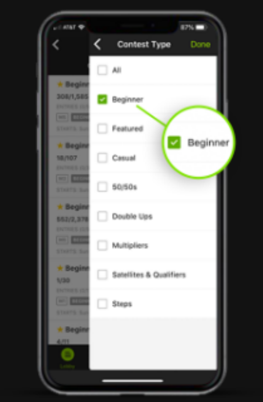
Who is eligible for beginner and casual Fantasy Sports contests? (US)

DraftKings is committed to maintaining a fair and level playing field to preserve the competitiveness of the game, which is why we offer different types of contests for players with different experience levels.

Beginner Contests

Beginner contests allow players new to DraftKings to start their career against other rookies. Beginner contests are open to less experienced players who haven't earned an experience badge and have played under 50 contests.

View the visual example below.



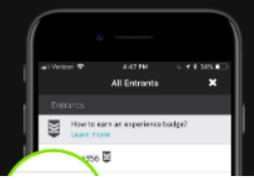
Casual Contests

Casual contests are open to players who have played in more than 50 contests but haven't earned an experience badge.

Experience Badges

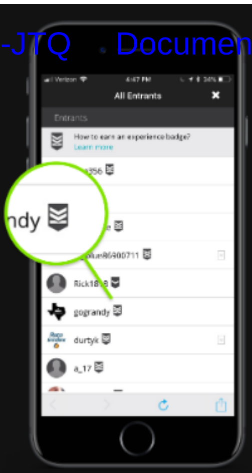
Experience badges signify contest entry milestones for accomplished players and allow other players to gauge their opponents' experience. **Anyone with an experience badge is ineligible to enter beginner and casual contests.**

View the visual example below.



JUMP TO SECTION
Fantasy Sports Basics
How to Play- Fantasy Sports
Game Styles & Contest Types
Contest Rules & Scoring
Resolving Fantasy Sports Issues
Tickets
Leagues

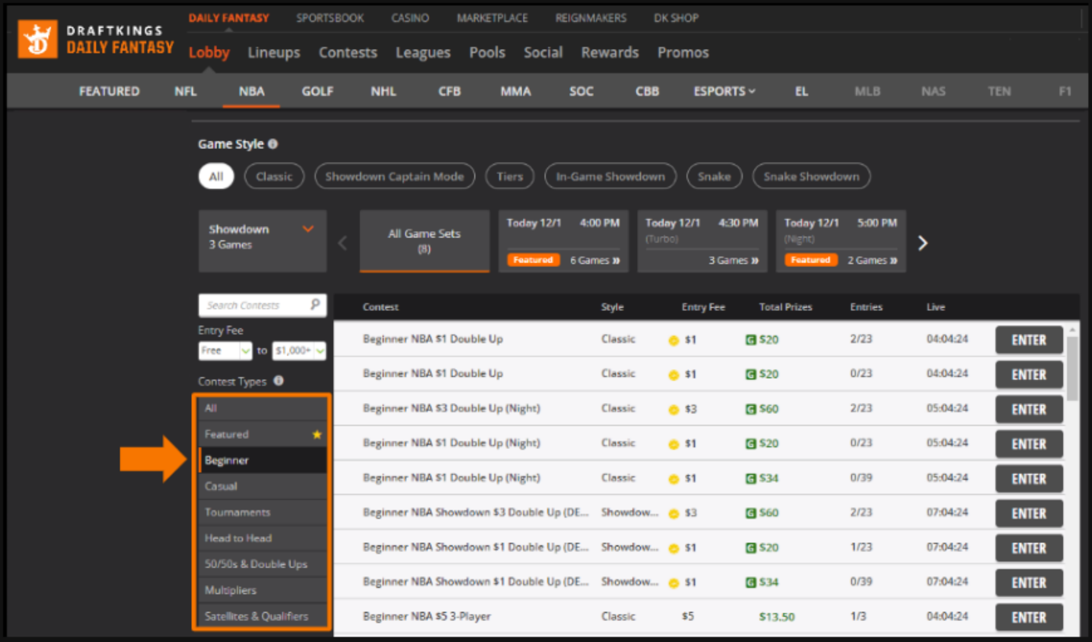
How can we help?



How to join either Beginner or Casual contests

1. Sign into your account and visit the **Lobby**.
2. Select your preferred **sport** from the menu.
3. Use the **Contest Types** filter in the Lobby to display **Beginner or Casual** contests only.

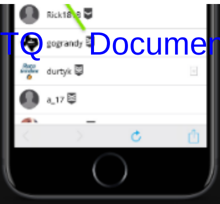
View the visual example below.



Contest	Style	Entry Fee	Total Prizes	Entries	Live	
Beginner NBA $1 Double Up	Classic	$1	$20	2/23	04:04:24	ENTER
Beginner NBA $1 Double Up	Classic	$1	$20	0/23	04:04:24	ENTER
Beginner NBA $3 Double Up (Night)	Classic	$3	$60	2/23	05:04:24	ENTER
Beginner NBA $1 Double Up (Night)	Classic	$1	$20	0/23	05:04:24	ENTER
Beginner NBA $1 Double Up (Night)	Classic	$1	$34	0/39	05:04:24	ENTER
Beginner NBA Showdown $3 Double Up (DE...	Showdow...	$3	$60	2/23	07:04:24	ENTER
Beginner NBA Showdown $1 Double Up (DE...	Showdow...	$1	$20	1/23	07:04:24	ENTER
Beginner NBA Showdown $1 Double Up (DE...	Showdow...	$1	$34	0/39	07:04:24	ENTER
Beginner NBA $5 3-Player	Classic	$5	$13.50	1/3	04:04:24	ENTER

Was this article helpful? Yes No

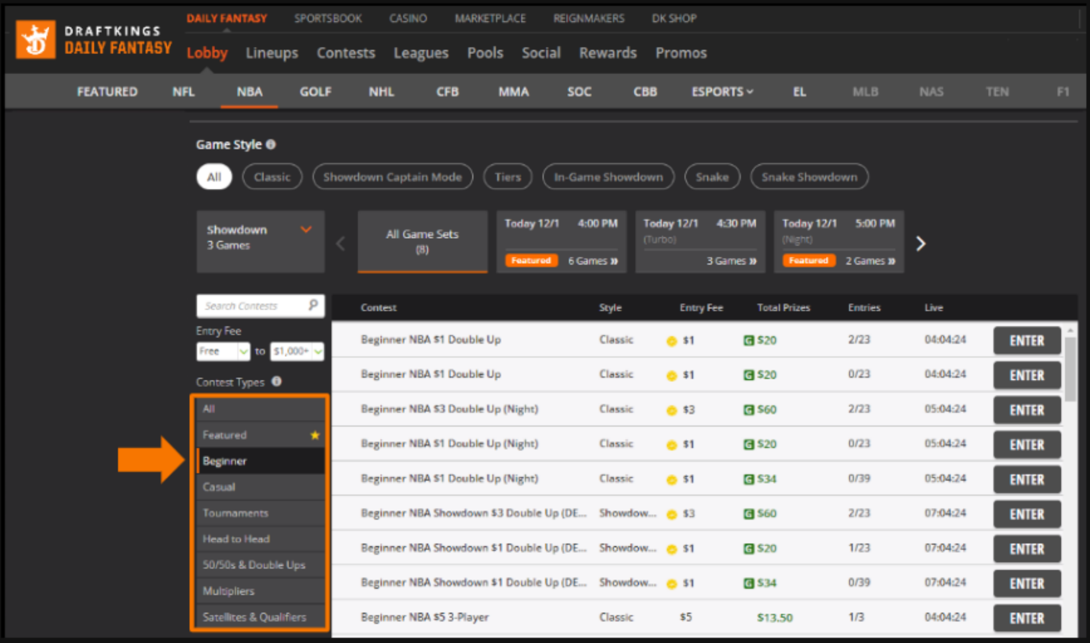
How can we help?



How to join either Beginner or Casual contests

1. Sign into your account and visit the **Lobby**.
2. Select your preferred **sport** from the menu.
3. Use the **Contest Types** filter in the Lobby to display **Beginner or Casual** contests only.

View the visual example below.


DRAFTKINGS DAILY FANTASY
DAILY FANTASY SPORTSBOOK CASINO MARKETPLACE REIGNMAKERS DK SHOP
Lobby Lineups Contests Leagues Pools Social Rewards Promos
FEATURED NFL NBA GOLF NHL CFB MMA SOC CBB ESPORTS EL MLB NAS TEN F1
Game Style
All
Classic
Showdown Captain Mode
Tiers
In-Game Showdown
Snake
Snake Showdown
Showdown 3 Games
All Game Sets (8)
Today 12/1 4:00 PM Featured 6 Games
Today 12/1 4:30 PM (Turbo) 3 Games
Today 12/1 5:00 PM (Night) Featured 2 Games
Search Contests
Entry Fee
Free to $1,000+
Contest Types
All
Featured
Beginner
Casual
Tournaments
Head to Head
50/50s & Double Ups
Multipliers
Satellites & Qualifiers

Contest	Style	Entry Fee	Total Prizes	Entries	Live	
Beginner NBA $1 Double Up	Classic	$1	$20	2/23	04:04:24	ENTER
Beginner NBA $1 Double Up	Classic	$1	$20	0/23	04:04:24	ENTER
Beginner NBA $3 Double Up (Night)	Classic	$3	$60	2/23	05:04:24	ENTER
Beginner NBA $1 Double Up (Night)	Classic	$1	$20	0/23	05:04:24	ENTER
Beginner NBA $1 Double Up (Night)	Classic	$1	$34	0/39	05:04:24	ENTER
Beginner NBA Showdown $3 Double Up (DE...	Showdow...	$3	$60	2/23	07:04:24	ENTER
Beginner NBA Showdown $1 Double Up (DE...	Showdow...	$1	$20	1/23	07:04:24	ENTER
Beginner NBA Showdown $1 Double Up (DE...	Showdow...	$1	$34	0/39	07:04:24	ENTER
Beginner NBA $5 3-Player	Classic	$5	$13.50	1/3	04:04:24	ENTER

Was this article helpful? Yes No

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

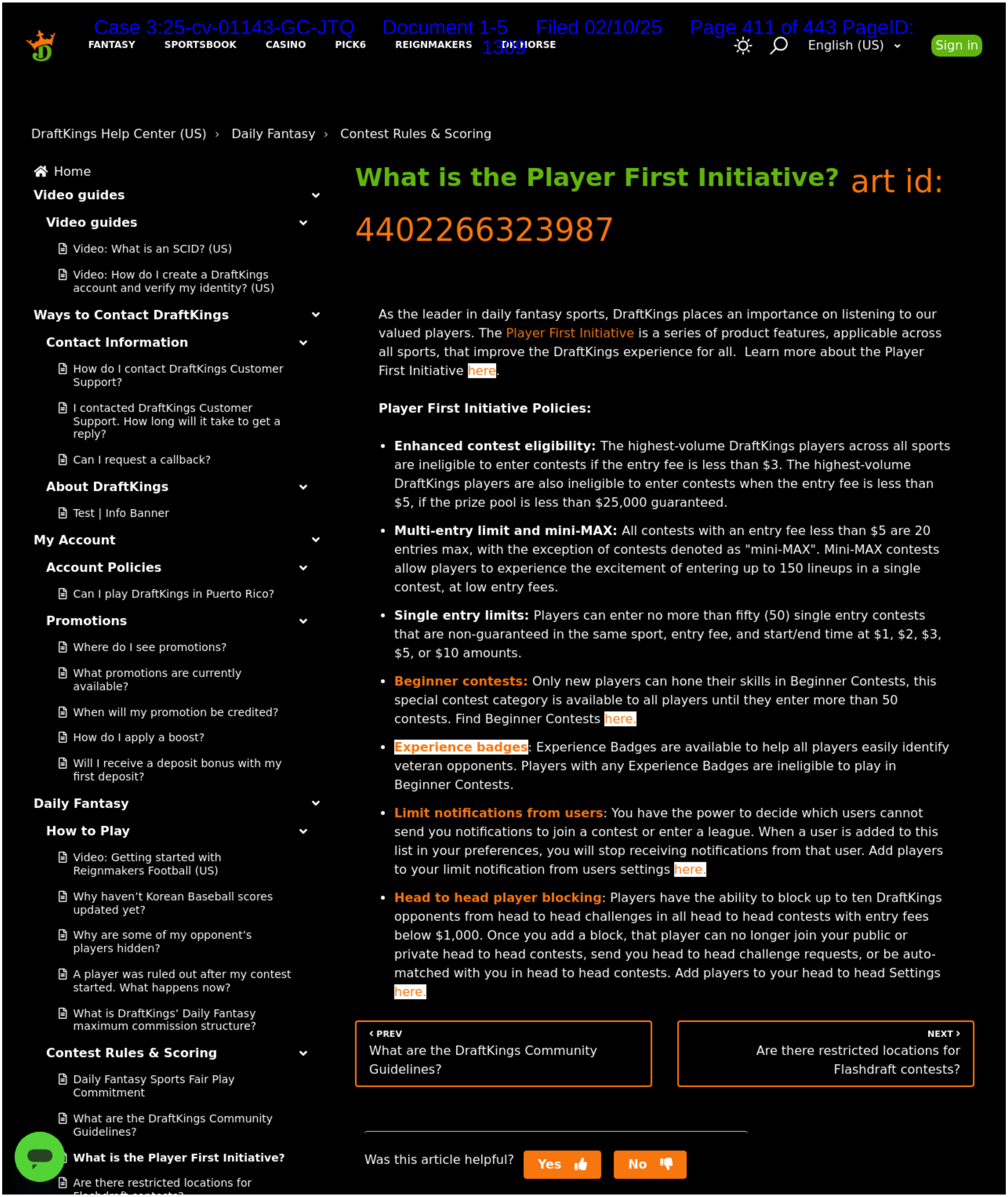
Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

How can we help?

Exhibit 68


FANTASY
SPORTSBOOK
CASINO
PICK6
REIGNMAKERS
DK HORSE
English (US)
Sign in
DraftKings Help Center (US) › Daily Fantasy › Contest Rules & Scoring
Home
Video guides
Video guides
Video: What is an SCID? (US)
Video: How do I create a DraftKings account and verify my identity? (US)
Ways to Contact DraftKings
Contact Information
How do I contact DraftKings Customer Support?
I contacted DraftKings Customer Support. How long will it take to get a reply?
Can I request a callback?
About DraftKings
Test | Info Banner
My Account
Account Policies
Can I play DraftKings in Puerto Rico?
Promotions
Where do I see promotions?
What promotions are currently available?
When will my promotion be credited?
How do I apply a boost?
Will I receive a deposit bonus with my first deposit?
Daily Fantasy
How to Play
Video: Getting started with Reignmakers Football (US)
Why haven't Korean Baseball scores updated yet?
Why are some of my opponent's players hidden?
A player was ruled out after my contest started. What happens now?
What is DraftKings' Daily Fantasy maximum commission structure?
Contest Rules & Scoring
Daily Fantasy Sports Fair Play Commitment
What are the DraftKings Community Guidelines?
What is the Player First Initiative?
Are there restricted locations for Flashdraft contests?

What is the Player First Initiative? art id: 4402266323987

As the leader in daily fantasy sports, DraftKings places an importance on listening to our valued players. The Player First Initiative is a series of product features, applicable across all sports, that improve the DraftKings experience for all. Learn more about the Player First Initiative here.

Player First Initiative Policies:

- **Enhanced contest eligibility:** The highest-volume DraftKings players across all sports are ineligible to enter contests if the entry fee is less than $3. The highest-volume DraftKings players are also ineligible to enter contests when the entry fee is less than $5, if the prize pool is less than $25,000 guaranteed.
- **Multi-entry limit and mini-MAX:** All contests with an entry fee less than $5 are 20 entries max, with the exception of contests denoted as "mini-MAX". Mini-MAX contests allow players to experience the excitement of entering up to 150 lineups in a single contest, at low entry fees.
- **Single entry limits:** Players can enter no more than fifty (50) single entry contests that are non-guaranteed in the same sport, entry fee, and start/end time at $1, $2, $3, $5, or $10 amounts.
- **Beginner contests:** Only new players can hone their skills in Beginner Contests, this special contest category is available to all players until they enter more than 50 contests. Find Beginner Contests here.
- **Experience badges**: Experience Badges are available to help all players easily identify veteran opponents. Players with any Experience Badges are ineligible to play in Beginner Contests.
- **Limit notifications from users**: You have the power to decide which users cannot send you notifications to join a contest or enter a league. When a user is added to this list in your preferences, you will stop receiving notifications from that user. Add players to your limit notification from users settings here.
- **Head to head player blocking**: Players have the ability to block up to ten DraftKings opponents from head to head challenges in all head to head contests with entry fees below $1,000. Once you add a block, that player can no longer join your public or private head to head contests, send you head to head challenge requests, or be auto-matched with you in head to head contests. Add players to your head to head Settings here.

‹ PREV
What are the DraftKings Community Guidelines?
NEXT ›
Are there restricted locations for Flashdraft contests?
Was this article helpful?
Yes
No

Document title: What is the Player First Initiative? – DraftKings Help Center (US)
Capture URL: https://draftkings1452613992.zendesk.com/hc/en-us/articles/4402266323987-What-is-the-Player-First-Initiative
Capture timestamp (UTC): Mon, 03 Feb 2025 22:29:42 GMT

- Daily Fantasy Sports Fair Play Commitment
- What are the DraftKings Community Guidelines?
- **What is the Player First Initiative?**
- Are there restricted locations for Flashdraft contests?
- Are there restricted locations for in-game contests?
- How do you determine the start time for In-Game Showdown contests?
- Can I edit the payout structure of a Daily Fantasy contest I created after it has been posted?
- What happens if I enter a multi-entry Daily Fantasy Sports contest and win in multiple places?
- Daily Fantasy Contest Rules & Scoring - overview
- How are Daily Fantasy contests scored?
- Can DraftKings employees play on DraftKings Daily Fantasy Sports and other DFS sites?
- Can employees of other DFS sites play on DraftKings?
- What happens when I tie with others in a Daily Fantasy Sports contest?

Contest Entry

- What types of wagers are available with DraftKings Sportsbook?
- How do I see the contests I've entered?
- What are experience badges?
- How do I play DraftKings Daily Fantasy (DFS)?
- I forgot to set my lineup. What can I do?
- How do I use my Crowns to enter into a contest?
- How do I reserve an entry?
- How does DraftKings treat late stat changes?

Tickets

- I won a ticket in a Satellite/Qualifier contest. How do I use it?

Leagues

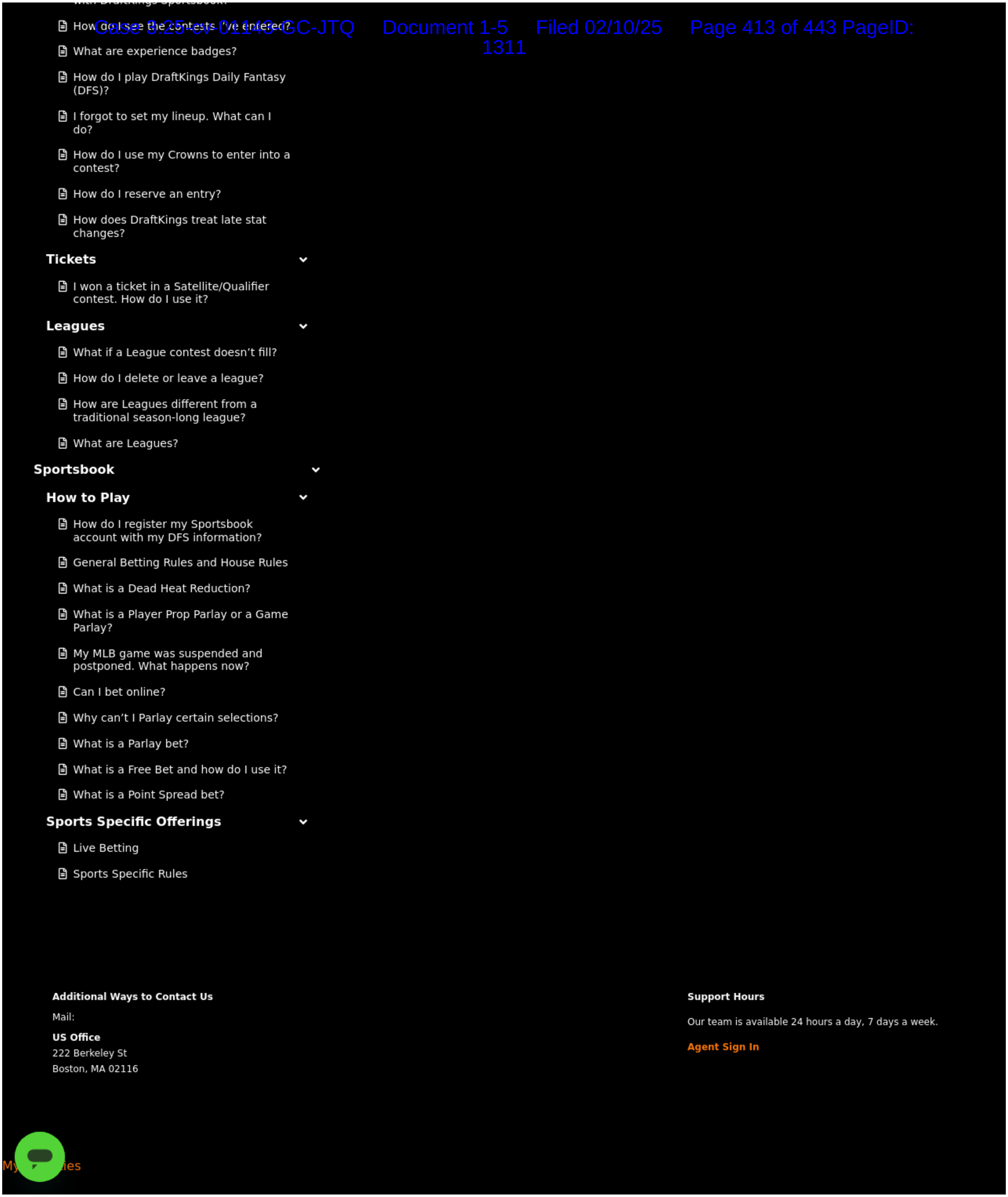
- What if a League contest doesn't fill?
- How do I delete or leave a league?
- How are Leagues different from a traditional season-long league?
- What are Leagues?

Sportsbook

How to Play

- How do I register my Sportsbook account with my DFS information?
- General Betting Rules and House Rules
- What is a Dead Heat Reduction?
- What is a Player Prop Parlay or a Game Parlay?

Was this article helpful? Yes No



with DraftKings Sportsbook?

- How do I see the contests I've entered?
- What are experience badges?
- How do I play DraftKings Daily Fantasy (DFS)?
- I forgot to set my lineup. What can I do?
- How do I use my Crowns to enter into a contest?
- How do I reserve an entry?
- How does DraftKings treat late stat changes?

Tickets

- I won a ticket in a Satellite/Qualifier contest. How do I use it?

Leagues

- What if a League contest doesn't fill?
- How do I delete or leave a league?
- How are Leagues different from a traditional season-long league?
- What are Leagues?

Sportsbook

How to Play

- How do I register my Sportsbook account with my DFS information?
- General Betting Rules and House Rules
- What is a Dead Heat Reduction?
- What is a Player Prop Parlay or a Game Parlay?
- My MLB game was suspended and postponed. What happens now?
- Can I bet online?
- Why can't I Parlay certain selections?
- What is a Parlay bet?
- What is a Free Bet and how do I use it?
- What is a Point Spread bet?

Sports Specific Offerings

- Live Betting
- Sports Specific Rules

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

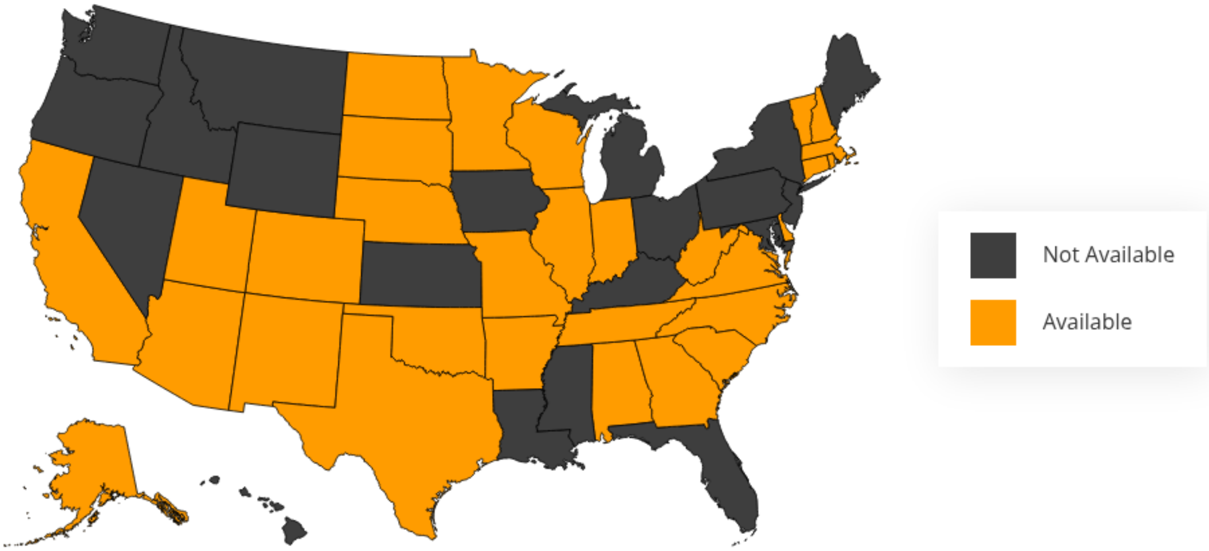
Our team is available 24 hours a day, 7 days a week.

Agent Sign In

My [illegible]ies

Exhibit 69

WHERE CAN YOU PLAY DRAFTKINGS PICK6?



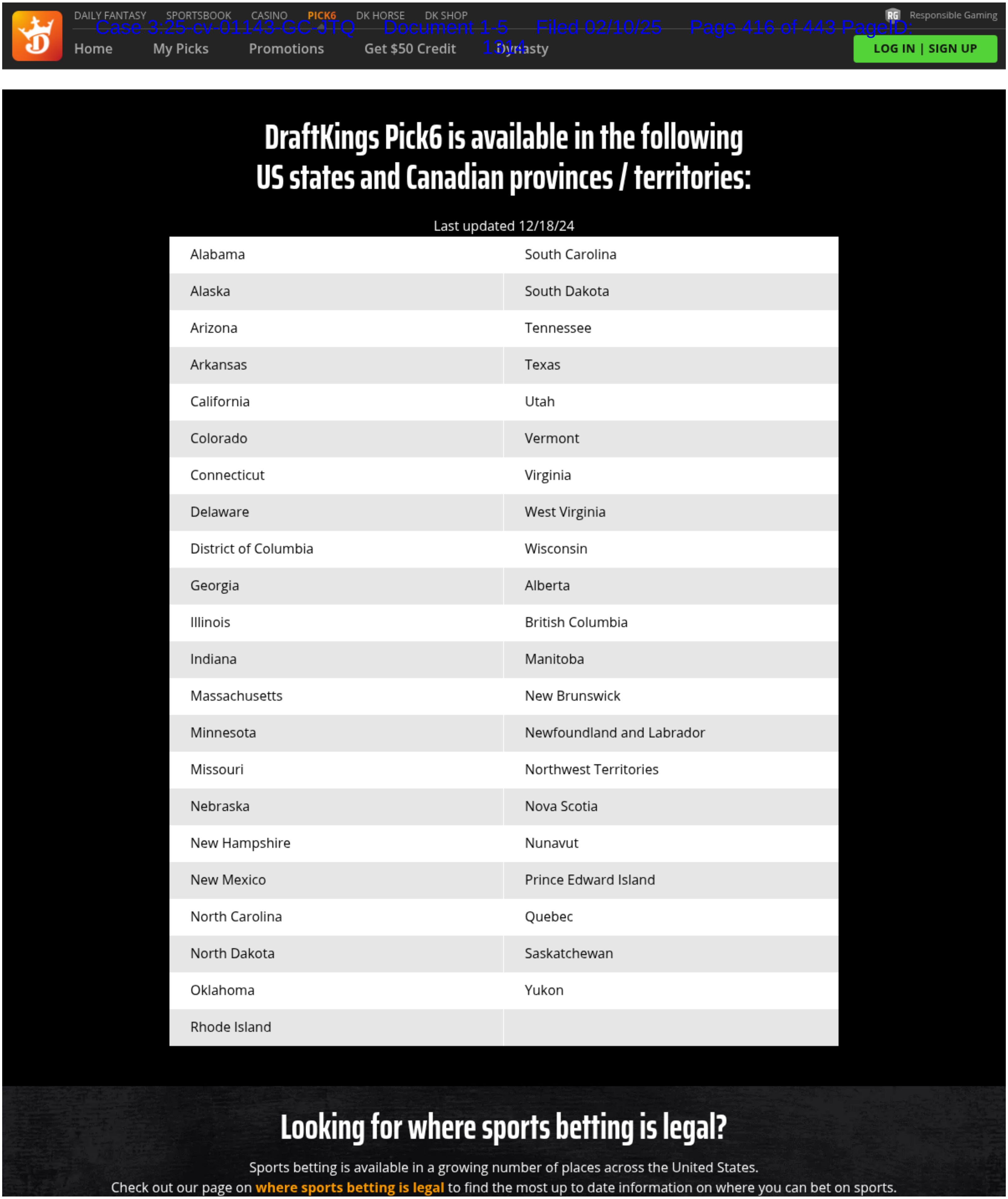
Start making player stat picks on DraftKings Pick6 today

PICK NOW

What is DraftKings Pick6?

Pick6 is a peer-to-peer fantasy game where you build a lineup (your “Pick Set”) of 2 or more athletes and select whether or not you believe each will outperform their listed stat projection. Enter your Pick Set in Pick6 contests to compete against other players. Get enough picks correct and win a share of cash prizes. You must be at least eighteen (18) years of age to open an account, participate in contests, or win prizes offered on Pick6. You must be at least nineteen (19) years of age at the time of account creation if you are a legal resident of Alabama or Nebraska, or at least twenty-one (21) years of age if you are a legal resident of Arizona, Iowa, Louisiana, or Massachusetts.

DAILY FANTASY SPORTSBOOK CASINO PICK6 DK HORSE DK SHOP Responsible Gaming



DraftKings Pick6 is available in the following US states and Canadian provinces / territories:

Last updated 12/18/24

Alabama	South Carolina
Alaska	South Dakota
Arizona	Tennessee
Arkansas	Texas
California	Utah
Colorado	Vermont
Connecticut	Virginia
Delaware	West Virginia
District of Columbia	Wisconsin
Georgia	Alberta
Illinois	British Columbia
Indiana	Manitoba
Massachusetts	New Brunswick
Minnesota	Newfoundland and Labrador
Missouri	Northwest Territories
Nebraska	Nova Scotia
New Hampshire	Nunavut
New Mexico	Prince Edward Island
North Carolina	Quebec
North Dakota	Saskatchewan
Oklahoma	Yukon
Rhode Island	

Looking for where sports betting is legal?

Sports betting is available in a growing number of places across the United States.
Check out our page on **where sports betting is legal** to find the most up to date information on where you can bet on sports.

Looking for where sports betting is legal?

Sports betting is available in a growing number of places across the United States.
Check out our page on **where sports betting is legal** to find the most up to date information on where you can bet on sports.

Looking for where online gambling is legal?

Online Gambling is available in a growing number of places across the United States. Check out our page on **where Online Gambling is legal** to find the most up-to-date information on where you can play.

We're bringing you action on all your devices

Download the DraftKings Pick6 app or play from your computer

Download on the App Store

GET IT ON Google Play

DraftKings
Boston, MA

COMPANY

- About DraftKings
- Terms of Use
- Fair Play Commitment
- Privacy Policy
- Responsible Gaming
- Average Results
- DK Nation Sports News
- DraftKings Sportsbook
- DraftKings Retail Sportsbook Locations
- DraftKings Casino
- DraftKings Daily Fantasy
- DraftKings Reignmakers
- DraftKings Mobile Apps
- Do Not Sell or Share My Personal Information

PICK6

- Pick6 NFL
- Pick6 NBA
- Pick6 NHL
- Pick6 PGA TOUR
- Pick6 UFC
- Pick6 College Football
- Pick6 College Basketball
- Pick6 Soccer
- Pick6 How to Play
- Pick6 Help Center (FAQs)
- Where is Pick6 Available?

LOCATIONS

US Office
222 Berkeley Street
Boston, MA
02116

Gambling Problem? Call 1-800-GAMBLER.

Must be at least 18 years or older. Higher age limits may apply in some states. Eligibility restrictions apply. Valid only in states where DraftKings Pick6 operates. Pick6 not available in all states, including, but not limited to, NY (for up-to-date list of states please visit pick6.draftkings.com/where-is-pick6-available). Void where prohibited. For entertainment purposes only. Winning a contest on DraftKings depends on knowledge and exercise of skill. See terms at pick6.draftkings.com.

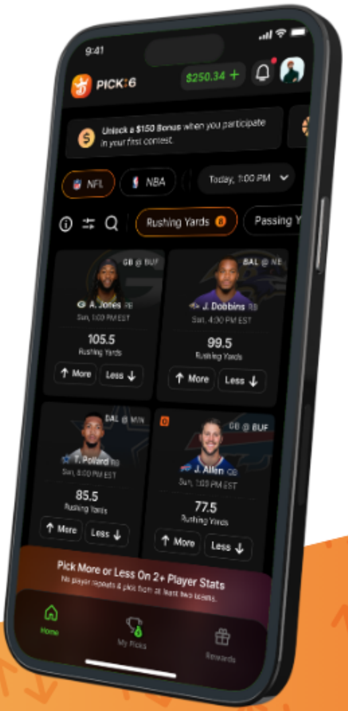
Exhibit 70

DRAFTKINGS PICK6

LEARN THE X'S AND O'S OF OUR NEWEST FANTASY GAME

Pick "more or less" on player projections. Compete for cash prizes. It's that simple.

Start Making Picks



Unlock a $150 Bonus when you participate in your first contest.

Pick More or Less On 2+ Player Stats

No player repeats & pick from at least two teams.

What is DraftKings Pick6?

Pick6 is a peer-to-peer fantasy game where you build an entry of 2+ players (3+ in Colorado) and select whether or not you believe each will outperform their listed stat projection. Once you craft your picks and choose your entry fee, your picks are entered into Pick6 contests to compete against those of other users. Get enough picks correct and win a share of huge cash prizes!

How to Play

Make your Picks

Build your entry of 2+ picks (3+ picks in Colorado). Your picks must contain athletes from the same sport and pick group (a set of picks available from a group of games/competitions). For each player, simply select if you think they'll have more or less than their listed stat projection. Please note: you can not pick the same player twice and you must pick players from at least two different teams.

Choose your Entry Fee

Once you finalize your picks, choose your entry fee amount. Your entries will be automatically distributed into available contests, subject to your confirmation. Get your

Choose your Entry Fee

Once you finalize your picks, choose your entry fee amount. Your entries will be automatically distributed into available contests, subject to your confirmation. Get your picks in today for as little as $1!

Sweat your Picks

When the games go live, use the My Picks tab to follow your picks in real time to track how your players are performing against their projections and how they stack up against other users. View your estimated prizing update in real time as picks in your contests are graded as wins or losses!

Compete for Prizes

Get enough correct picks and win a share of the contests' guaranteed prizes. Any prizes won will be credited to your DraftKings account after contests are finalized. View our **Pick6 Prizes page** to see what customers have won in recent days!

Sports you can play

COLLEGE BASKETBALL

Make your NBA picks on points, assists, and more to score cash prizes!

Play Now

College Hoops is now live on Pick6! Location restrictions apply.

Play Now

NFL picks are back! Compete for huge cash prizes on the gridiron.

Play Now

Submit your CFB picks today! Location restrictions apply.

Play Now

New sport! Make your picks for projections on the pitch!

Play Now

Pick6 Promotions

Don't miss out on our latest promotions, exclusive offers, and bonuses.

See Promotions

Pick6 Promotions

Don't miss out on our latest promotions, exclusive offers, and bonuses.

See Promotions

We're bringing you action on all your devices

Download the DraftKings Pick6 app or play from your computer

Download on the App Store

GET IT ON Google Play

Pick6 FAQs

How do I access Pick6? ▾

Where is Pick6 available? ▾

Which stat projections are available in Pick6? ▾

Where can I learn more about how Pick6 works? ▾

DraftKings
Boston, MA

COMPANY

- About DraftKings
- Terms of Use
- Fair Play Commitment
- Privacy Policy
- Responsible Gaming
- Average Results
- DK Nation Sports News
- DraftKings Sportsbook
- DraftKings Retail Sportsbook Locations
- DraftKings Casino

PICK6

- Pick6 NFL
- Pick6 NBA
- Pick6 NHL
- Pick6 PGA TOUR
- Pick6 UFC
- Pick6 College Football
- Pick6 College Basketball
- Pick6 Soccer
- Pick6 How to Play
- Pick6 Help Center (FAQs)

LOCATIONS

US Office
222 Berkeley Street
Boston, MA
02116

Pick6 FAQs

How do I access Pick6? ▾

Where is Pick6 available? ▾

Which stat projections are available in Pick6? ▾

Where can I learn more about how Pick6 works? ▾

DraftKings
Boston, MA

COMPANY

- About DraftKings
- Terms of Use
- Fair Play Commitment
- Privacy Policy
- Responsible Gaming
- Average Results
- DK Nation Sports News
- DraftKings Sportsbook
- DraftKings Retail Sportsbook Locations
- DraftKings Casino
- DraftKings Daily Fantasy
- DraftKings Reignmakers
- DraftKings Mobile Apps
- Do Not Sell or Share My Personal Information

PICK6

- Pick6 NFL
- Pick6 NBA
- Pick6 NHL
- Pick6 PGA TOUR
- Pick6 UFC
- Pick6 College Football
- Pick6 College Basketball
- Pick6 Soccer
- Pick6 How to Play
- Pick6 Help Center (FAQs)
- Where is Pick6 Available?

LOCATIONS

US Office
222 Berkeley Street
Boston, MA
02116

Gambling Problem? Call 1-800-GAMBLER.

Must be at least 18 years or older. Higher age limits may apply in some states. Eligibility restrictions apply. Valid only in states where DraftKings Pick6 operates. Pick6 not available in all states, including, but not limited to, NY (for up-to-date list of states please visit pick6.draftkings.com/where-is-pick6-available). Void where prohibited. For entertainment purposes only. Winning a contest on DraftKings depends on knowledge and exercise of skill. See terms at pick6.draftkings.com.

Exhibit 71

DRAFTKINGS PICK6 REVIEW: WHERE IS DRAFTKINGS PICK 6 LEGAL + AGE TO WAGER ON THE DRAFTKINGS PICK6 APP

WHAT IS DRAFTKINGS PICK 6?

3.8/5 ★★★★★

Get to know DraftKings Pick6 as part of this comprehensive resource. We've compiled everything there is to know about this awesome pick-and-play fantasy variant, so you can get up to speed and start competing for real cash prizes.

Pick6 from DK is available in 20 US states including territories with no other legal sports betting. This, coupled with the DraftKings $200 in Pick6 credits if your first ever pick set doesn't win, makes the fantasy sports system a dream. Find out how to get started with DK Pick6 right here.

DETAILS | PAYMENT METHODS | USER REVIEWS | HOW WE RATE

PAYMENT METHODS	ACCEPTS CUSTOMERS FROM	WITHDRAWAL	COMPANY
PayPal, Play+, preferred e-check	Show States	1 – 5 Days	DraftKings Ltd

WEBSITE	AVAILABLE LANGUAGES	CUSTOMER SUPPORT	EMAIL
pick6.draftkings.com	EN	Phone, Email Support	support@draftkings.com

PHONE

877-659-6066

David Astramskas (Redapples) EDITOR IN CHIEF
Last Updated on 02/03/2025

Fact checked by: Sara Jane Gamelli

Table of Contents

1. Your Pick6 DraftKings guide
2. Pick6 DraftKings FAQs answered
3. Our final say on the Pick6 fantasy system from DraftKings
4. DraftKings Pick6 FAQs

YOUR PICK6 DRAFTKINGS GUIDE

WHAT IS DRAFTKINGS PICK6?

Pick6 is an innovative new fantasy pick-and-play system that lets you compete with real opponents in a peer-to-peer gaming environment. Participants must pick 2-6 players to create a "Pick Set", with the goal being to outperform their opponents based on the stat projections of their selections.

Pick6 is an innovative new fantasy pick-and-play system that lets you compete with real opponents in a peer-to-peer gaming environment. Participants must pick 2-6 players to create a "Pick Set", with the goal being to outperform their opponents based on the stat projections of their selections.

We'll cover the ins and outs of how to play DraftKings Pick6 later in the guide, but suffice it to say that this is a truly unique sports betting experience. Currently, you can make Pick6 selections on the NFL, NHL and NBA with more sports set to be integrated into future iterations of the game. DK Pick6 is available on desktop and mobile devices, with an intuitive smartphone app facilitating convenient on-the-go play.

HOW TO PLAY DRAFTKINGS PICK6

It's time to take you step-by-step through the process of playing Pick6 at DraftKings. If you've previously competed in online fantasy leagues some elements of Pick6 DK should be pretty familiar, although there are certain to be some unique features and additions that you may never have encountered before. After all, this is a totally unique fantasy variant that has brought some pioneering new features to the fore.

With all that said, let's take a closer look at how the DK Pick6 system works.

Step 1: Register for a new DraftKings Pick6 account

Since Pick6 is a DraftKings product, you can use your existing DK login credentials to get started with the brand's new pick-and-play fantasy system. If that applies to you, jump ahead to the next step.

Not yet registered with DraftKings? No sweat. As covered in our full DraftKings review, getting started with this awesome US sports betting specialist couldn't be simpler and you can join directly from the Pick6 portal.

To register with DraftKings, simply:

1. Tap the green "Log in/Sign Up" button in the top right of the main DK Pick6 play screen. Note that you can also create a new DK account directly in the Pick6 app.
2. Enter the required details including your preferred email, username, and password, before adding your phone number. There's also the option to add promo codes at this stage, so be sure to check out our guide to the best DraftKings Pick6 promos before you start.
3. The final registration step is to read the T&Cs and agree to the DK terms of use. Note that players must be 18+ to access all DK Fantasy products, rising to 19+ and 21+ in selected jurisdictions (see my list of DK Pick6 legal states below for more information).

All registered? Great – then it's time to play Pick6 using the Draftkings Bonus.

Step 2: Create a pick set

When you land in your shiny new DraftKings Pick6 account, your first order of business will be to create your first "Pick Set" – the team of players from the NBA, NFL, or NHL that you want to make a part of your first fantasy contest.

A Pick Set at Pick6 DK consists of 2-6 players from the same sport. For each player you choose, you'll need to predict whether they have more or less than their stat projection. Players can't be chosen twice on any Pick Set and you must choose players from two teams as a minimum.

Remember: since DK Pick6 is a peer-to-peer fantasy league, your Pick Sets will be pitted against other participants within contests. As well as being super fun and competitive, this also brings peace of mind that the game environment is fair and unbiased.

Remember: since DK Pick6 is a peer-to-peer fantasy league, your Pick Set will be pitted against other participants within contests. As well as being super fun and competitive, this also brings peace of mind that the game environment is fair and unbiased.

Step 3: Choose your bet amount

Once you've tweaked your Pick Set to the nth degree, the next step is to choose the entry bet amount you want to play with. Note that however much you choose to play with, the amount will be spread proportionately across all available contests; you can customize this spread manually before confirming your final bet amounts.

Step 4: Monitor your Pick Set to gauge how it's performing

When games get underway, both the DraftKings Pick6 app and the desktop site allow you to monitor the progress of your selections to see how they're performing against other participants in a given contest prize pool. To see a live view of your Pick Set based on current gameplay, simply visit the My Picks tab on the DraftKings Pick6 site or in the app.

Step 5: Pick up your winnings

If it turns out that you have a natural aptitude for DK Pick6 and manage to place enough correct picks to earn a share of a contest's prize pot, your prize will be automatically credited to your DraftKings account once all live games and contests are finalized

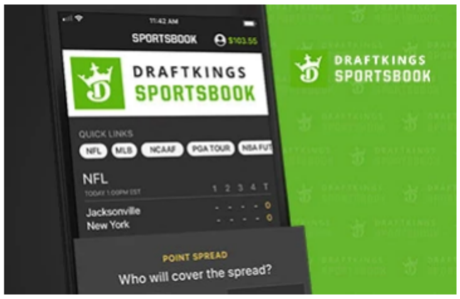
Typically, Draftkings Pick6 payout amounts are based on the following criteria:

- The number of correct picks made in a qualifying Pick Set
- The number of "shares" you had in a given contest

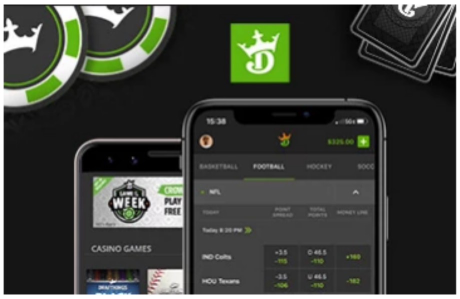
As an example, let's say a 5-Pick NBA contest has a $10,000 prize pool. Pick Sets with 5/5 correct selections would take home $8,000, while those with 4/5 would bank $2,000.

All DraftKings Pick6 contests feature an estimated prize pot based on individual wagering amounts, which you can review prior to entry.

And that's basically the long and short of how Pick6 by DraftKings works. In my experience, the more you experiment, tweak, and play around with this awesome sports selector game, the more you'll ultimately glean the best strategies and tactics to use. So, don't be afraid to get creative while keeping those early bets small, manageable, and in line with your regular gambling habits.



DraftKings Pick6 offers



Bet from your mobile at DraftKings

PICK6 DRAFTKINGS FAQS ANSWERED

I know there's a lot to get your head around when starting out with DK Pick6 as a complete beginner, but I promise it will be totally worth the effort. After all, this is a thrill-a-minute sports gaming platform that takes fantasy league play to a whole new level.

So, aside from the specific gameplay rules associated with Pick6, what else do you need to know? To answer that, we've handpicked a bunch of Pick6-related FAQs that we'll be answering in detail below.

that takes fantasy league play to a whole new level.

So, aside from the specific gameplay rules associated with Pick6, what else do you need to know? To answer that, we've handpicked a bunch of Pick6-related FAQs that we'll be answering in detail below.

WHAT STATES IS DRAFTKINGS PICK6 LEGAL IN?

I promised to cover the full list of DraftKings Pick6 legal states so here you go. The table below shows a list of all the places where you can legally use Pick6 in the US, along with their required age limits.

DK legal state	DK Pick6 age limit
Alaska	19+
Arizona	21+
California	18+
Georgia	18+
Iowa	21+
Louisiana	21+
Massachusetts	21+
Minnesota	18+
Missouri	18+
Nebraska	19+
New Mexico	18+
North Dakota	18+
Oklahoma	18+
South Carolina	18+
South Dakota	18+
Tennessee	18+
Utah	18+
Wisconsin	18+
West Virginia	18+
Washington DC	18+

Not lucky enough to live in a state where Pick6 DK is currently available? Don't worry – DraftKings is working on bringing its pioneering new fantasy variant to other parts of the US, so be sure to follow the brand's socials for updates on incoming expansions.

WHAT SPORTS ARE COVERED BY DK PICK6?

When making your Pick Set selections, you'll have a choice of three pro-league sports markets to choose from, including:

- ✓ DraftKings NBA Pick6
- ✓ DraftKings NFL Pick6

choose from, including:

- ✓ DraftKings NBA Pick6
- ✓ DraftKings NFL Pick6
- ✓ DraftKings NHL Pick6

This may not seem like a very flexible offering, but given the teams, contests, players, and matches on offer, you should find plenty of unique ways to come up with winning Pick Sets. Plus, you need to bear in mind that the DraftKings Pick6 platform is still relatively new, so expect to see new sports markets added in the future.

WHAT ARE THE BEST STRATEGIES FOR PICK6 ON DRAFTKINGS?

I've covered our recommended Draftkings pick strategy in full as part of this linked-to resource, so be sure to check out my top tips if you're looking to out-manoeuvre your opponents and take your pick-selection tactics up a gear.

There are multiple strategies that players can adopt at Pick6 but, of course, none of these are a complete guarantee of success. But it almost goes without saying that if you're a basketball fan, then the DraftKings NBA Pick6 is the right choice for you, as regular research into player performance and team tactics is essential for assembling a winning squad for your Pick6 plays.

ARE THERE ANY WELCOME BONUSES ON OFFER FOR JOINING PICK6 AT DRAFTKINGS?

Yes, there's actually a very nice welcome bonus available. New customers that join DraftKings specifically to play in Pick6 fantasy contests can earn up to $200 in Pick6 Credits if their first ever pick set doesn't win. There are, of course, a couple of T&Cs that you need to be aware of with regards to the Pick6 welcome offer, including:

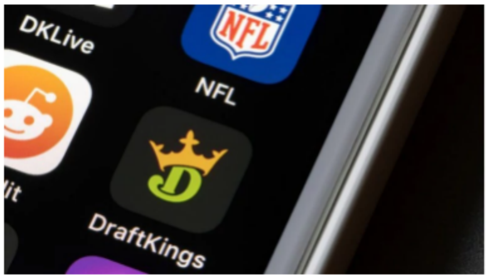
- ✓ You must meet the $5 minimum threshold on a single Pick Set
- ✓ You must place your entry within promotional period for it to count
- ✓ A 1x playthrough requirement applies to any winnings earned in conjunction with the bonus

CAN I REFER A FRIEND TO PICK6?

Players who successfully use the Draftkings Pick6 refer a friend scheme to onboard an associate onto the Pick6 platform can receive up to $50 in free bet credit. Any players you introduce Pick6 to must never have held a DraftKings account before, and they must first make a deposit of over $25 before any bonus sums are transferred.

WHAT ARE THE PROS AND CONS OF DK PICK6?

Not yet convinced that Pick6 from DK is right for you? We've pulled together the game's essential pros and cons so you can decide if it's a good fit for you.



Pros & Cons

- ✓ Awesome cash prize pools available
- ✓ Strategy-led gameplay that requires tactical prowess
- ✓ Top sports betting promotions available
- ✓ Download the DK Pick6 app or play on your computer

- ● Steep learning curve for some players

OUR FINAL SAY ON THE PICK6 FANTASY SYSTEM FROM DRAFTKINGS

As one of the best US betting sites, it's no surprise to find that DraftKings is also responsible for what is arguably one of the most exciting evolutions in sports betting since the legalization of online sportsbooks. Whether you've previously participated in fantasy pick-and-play games or you are a total beginner, I promise that a thrill-a-minute experience awaits here.

As one of the best US betting sites, it's no surprise to find that DraftKings is also responsible for what is arguably one of the most exciting evolutions in sports betting since the legalization of online sportsbooks. Whether you've previously participated in fantasy pick-and-play games or you are a total beginner, I promise that a thrill-a-minute experience awaits here.

Of course, there are a couple of niggles here, including Pick6's relatively limited availability across the US. But with DK promising new sports markets and territory expansions in the future, I hope these slight bugbears won't prove a dealbreaker for you.

In the meantime, if you are lucky enough to live in a state with access to DK Pick6, I'd urge you to give it a go. And with up to $200 in Pick6 Credits currently available thanks to the DraftKings welcome bonus, there's never been a better time to play.

DRAFTKINGS PICK6 FAQS

1 🔎 IS PICK6 LEGAL IN THE US?

Pick6 from DraftKings is legal in 20 US states, so be sure to check this guide for a full list of legal territories and age restrictions.

2 🎁 IS THERE A WELCOME BONUS WHEN JOINING DK PICK6?

Yes, new players who have never had a DraftKings account before can earn up to $200 in Pick6 credits if their first ever pick set doesn't win.

3 👀 WILL DRAFTKINGS ADD MORE SPORTS TO PICK6?

Yes, DraftKings intends to add more sports to its new but in-demand Pick6 product in the future. Currently, you can choose players from DraftKings NFL Pick6, or select your own NBA and NHL teams.

USER REVIEWS FOR DRAFTKINGS PICK6

No review posted yet.

Write a Review ›

POST YOUR OWN REVIEW

Name *

Email (not public) *

Your Rating: * ★★★★★

Title of your Review *

What's positive?

Title of your Review *

What's positive?

What's negative?

Submit Review

* required

BALLISLIFE.COM MOST VISITED

- Ballislife Videos
- Ballislife News
- Ballislife Profiles
- Ballislife Fab 50
- Ballislife Podcasts
- Ballislife Betting News

OUR TEAM

+5

Sara Jane Gamelli
Writer | Bettor

Sadonna Price
Editor | Writer

John Carlo A. Villaruel
Sports Writer

Kenny Ducey
Sports Reporter | Capper

Ronnie Flores
Editor | Senior Writer

David Astramskas (Redapples)
Editor | Senior Writer

Joseph Scalise
Editor | Writer

MORE ARTICLES RELATED TO DRAFTKINGS PICK6

→ Draftkings Pick6 Promo Code: Claim Excl...

Top Brands | Top Bonuses

UNDERDOG FANTASY	**Underdog Fantasy** Underdog Fantasy Review	4.6/5 ★★★★★
parlay play	**ParlayPlay** ParlayPlay Review	4.6/5 ★★★★★
dabble	**Dabble** Dabble Review	4.5/5 ★★★★★

Dabble	Dabble Review	4.5/5 ★★★★★
OwnersBox	OwnersBox Review	4.5/5 ★★★★★
DraftKings Fantasy	DraftKings Fantasy Review	4.5/5 ★★★★★

Go to DFS Comparison

TOP DAILY FANTASY SPORTS GUIDES

- DFS Sites: Best 5 DFS Apps
- NBA DFS: Top 5 NBA DFS Apps
- NCAAB DFS: Our Top 5 DFS Apps for NCAA

ABOUT BALLISLIFE

© 2005-2024
BALLISLIFE.COM -
PO BOX 15355.
IRVINE, CA 92623

- About Us
- Authors
- Contact Us
- Terms of Use
- DMCA

BallIsLife.com is available in the following countries:

US English (US)

MORE ABOUT US

- About Ballislife
- The Ballislife Team
- Content Enquiries
- How We Rate
- Content Disclaimer
- Affiliate Disclaimer
- Responsible Gambling
- Writers Wanted
- Sitemap
- Post Sitemap
- News Sitemap
- Ontario Sitemap

TRENDING ARTICLES

- Underdog Fantasy Legal States
- Pulsz Redemption
- Fliff Coins To Cash
- Fliff Legal States
- Fliff Review
- Stake.US Free Spins
- Stake.US VIP Bonuses
- Stake Free 7 Dollars
- How To Deposit on Stake
- Stake US Sweepstakes
- Is Stake Casino Legal in US?
- How To Buy Gold Coins On Stake.US

LATEST ARTICLES

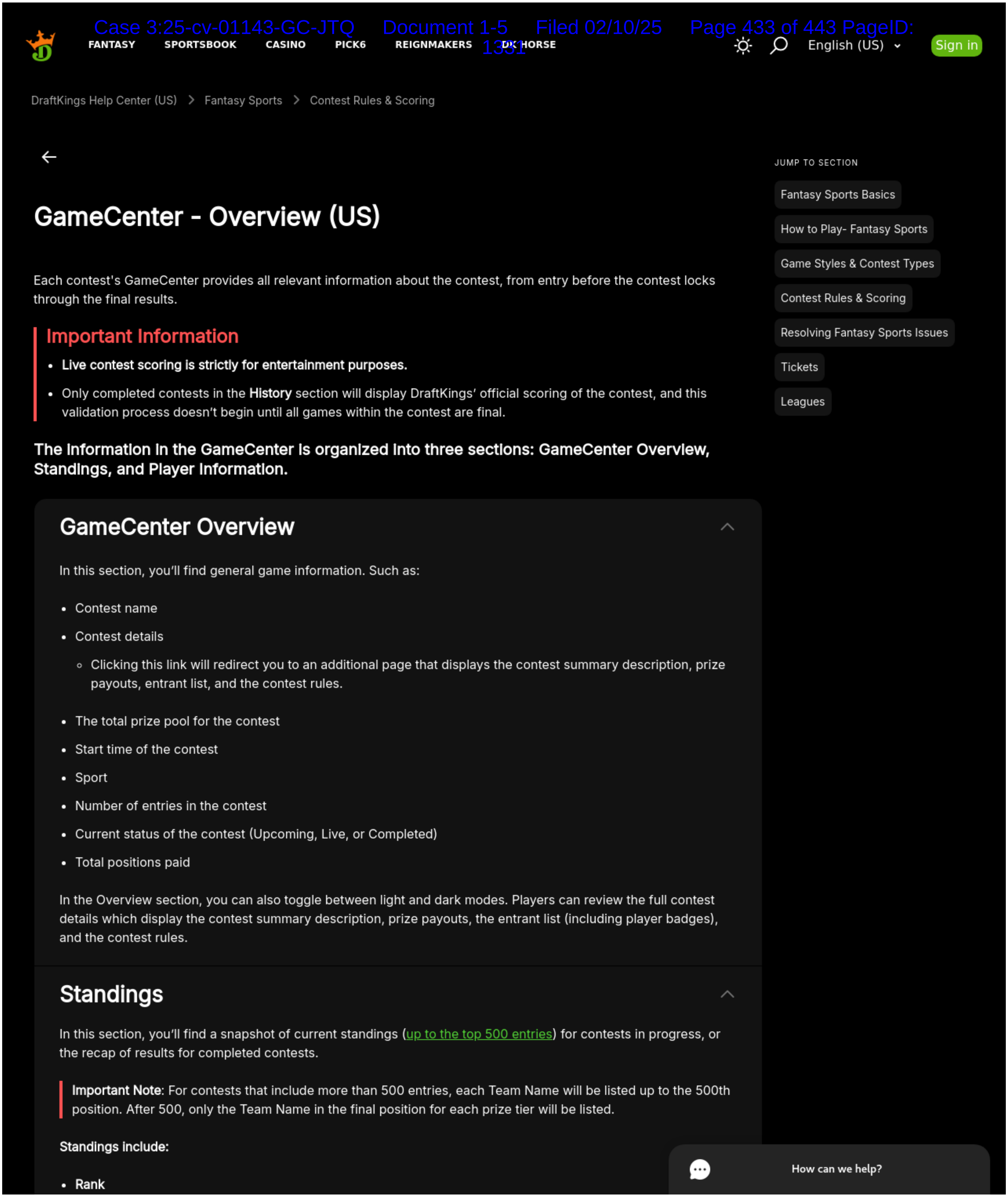
- Bet365 Referral Code
- Underdog $500 Match
- Stake.US Referral Code
- Stake.US Daily Reload
- Apps Like Fliff
- Fliff Referral Code
- BetMGM Refer a Friend
- Stake Free Money Code
- Can You Win Real Money at Pulsz?
- Stake.US Weekly Bonus Code
- BetMGM Bonus Code For Existing Users

Players must be 21 years of age or older or reach the minimum age for gambling in their respective state and located in jurisdictions where online gambling is legal. Please play responsibly. Bet with your head, not over it. If you or someone you know has a gambling problem, and wants help, call or visit: (a) the Council on Compulsive Gambling of New Jersey at 1-800-Gambler or www.800gambler.org; or (b) Gamblers Anonymous at 855-2-CALL-GA or www.gamblersanonymous.org.

This site is using Cloudflare and adheres to the Google Safe Browsing Program. We adapted Google's Privacy Guidelines to keep your data safe at all times.

21+ NCPG NCPG GA SECURE

Exhibit 72


FANTASY
SPORTSBOOK
CASINO
PICK6
REIGNMAKERS
DK HORSE
English (US)
Sign in
DraftKings Help Center (US) > Fantasy Sports > Contest Rules & Scoring
JUMP TO SECTION
Fantasy Sports Basics
How to Play- Fantasy Sports
Game Styles & Contest Types
Contest Rules & Scoring
Resolving Fantasy Sports Issues
Tickets
Leagues
How can we help?

GameCenter - Overview (US)

Each contest's GameCenter provides all relevant information about the contest, from entry before the contest locks through the final results.

Important Information

- **Live contest scoring is strictly for entertainment purposes.**
- Only completed contests in the **History** section will display DraftKings' official scoring of the contest, and this validation process doesn't begin until all games within the contest are final.

The information in the GameCenter is organized into three sections: GameCenter Overview, Standings, and Player Information.

GameCenter Overview

In this section, you'll find general game information. Such as:

- Contest name
- Contest details
  - Clicking this link will redirect you to an additional page that displays the contest summary description, prize payouts, entrant list, and the contest rules.
- The total prize pool for the contest
- Start time of the contest
- Sport
- Number of entries in the contest
- Current status of the contest (Upcoming, Live, or Completed)
- Total positions paid

In the Overview section, you can also toggle between light and dark modes. Players can review the full contest details which display the contest summary description, prize payouts, the entrant list (including player badges), and the contest rules.

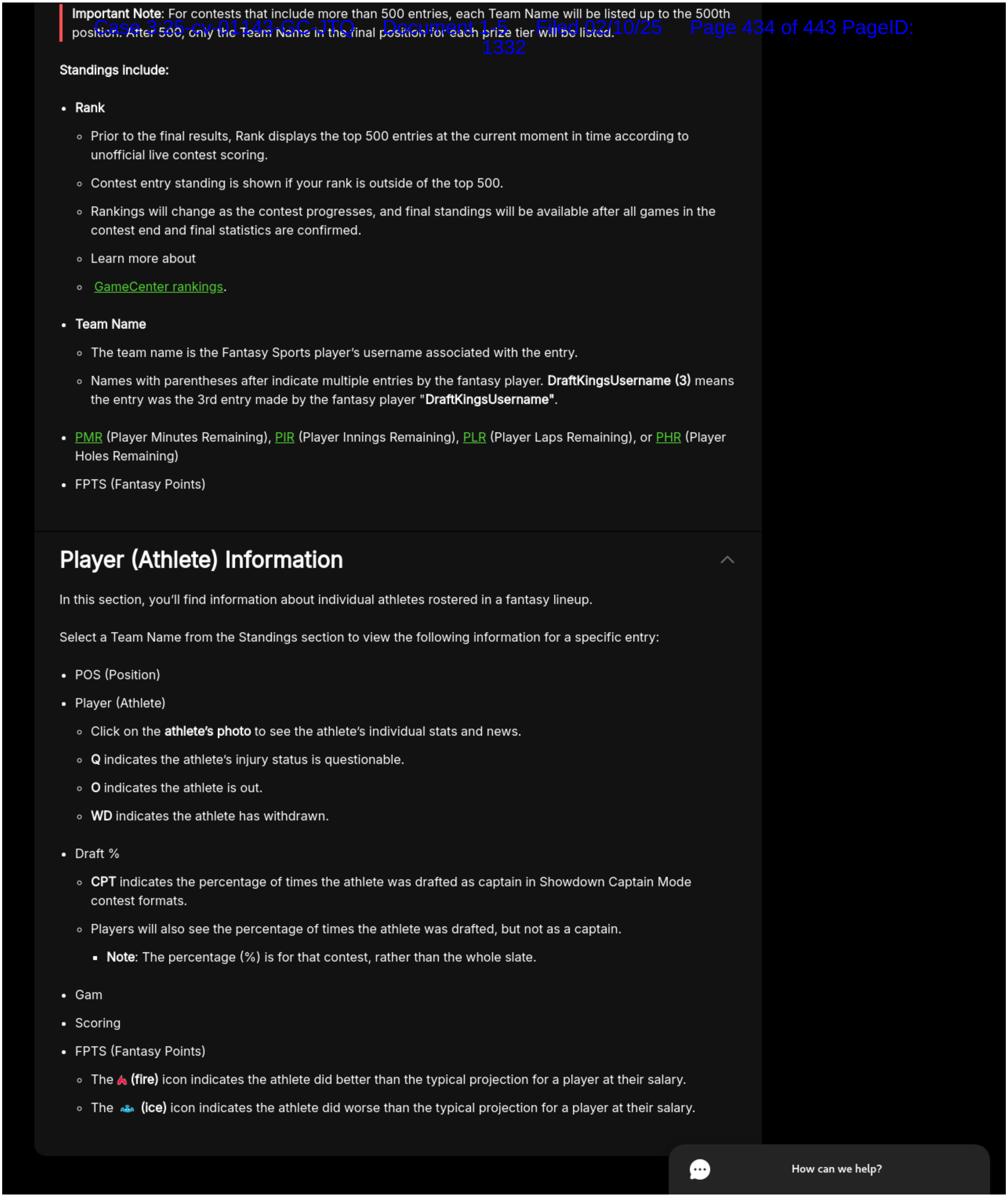
Standings

In this section, you'll find a snapshot of current standings (up to the top 500 entries) for contests in progress, or the recap of results for completed contests.

> **Important Note**: For contests that include more than 500 entries, each Team Name will be listed up to the 500th position. After 500, only the Team Name in the final position for each prize tier will be listed.

Standings include:

- Rank



Important Note: For contests that include more than 500 entries, each Team Name will be listed up to the 500th position. After 500, only the Team Name in the final position for each prize tier will be listed.

Standings include:

- **Rank**
  - Prior to the final results, Rank displays the top 500 entries at the current moment in time according to unofficial live contest scoring.
  - Contest entry standing is shown if your rank is outside of the top 500.
  - Rankings will change as the contest progresses, and final standings will be available after all games in the contest end and final statistics are confirmed.
  - Learn more about
  - GameCenter rankings.
- **Team Name**
  - The team name is the Fantasy Sports player's username associated with the entry.
  - Names with parentheses after indicate multiple entries by the fantasy player. **DraftKingsUsername (3)** means the entry was the 3rd entry made by the fantasy player "**DraftKingsUsername"**.
- PMR (Player Minutes Remaining), PIR (Player Innings Remaining), PLR (Player Laps Remaining), or PHR (Player Holes Remaining)
- FPTS (Fantasy Points)

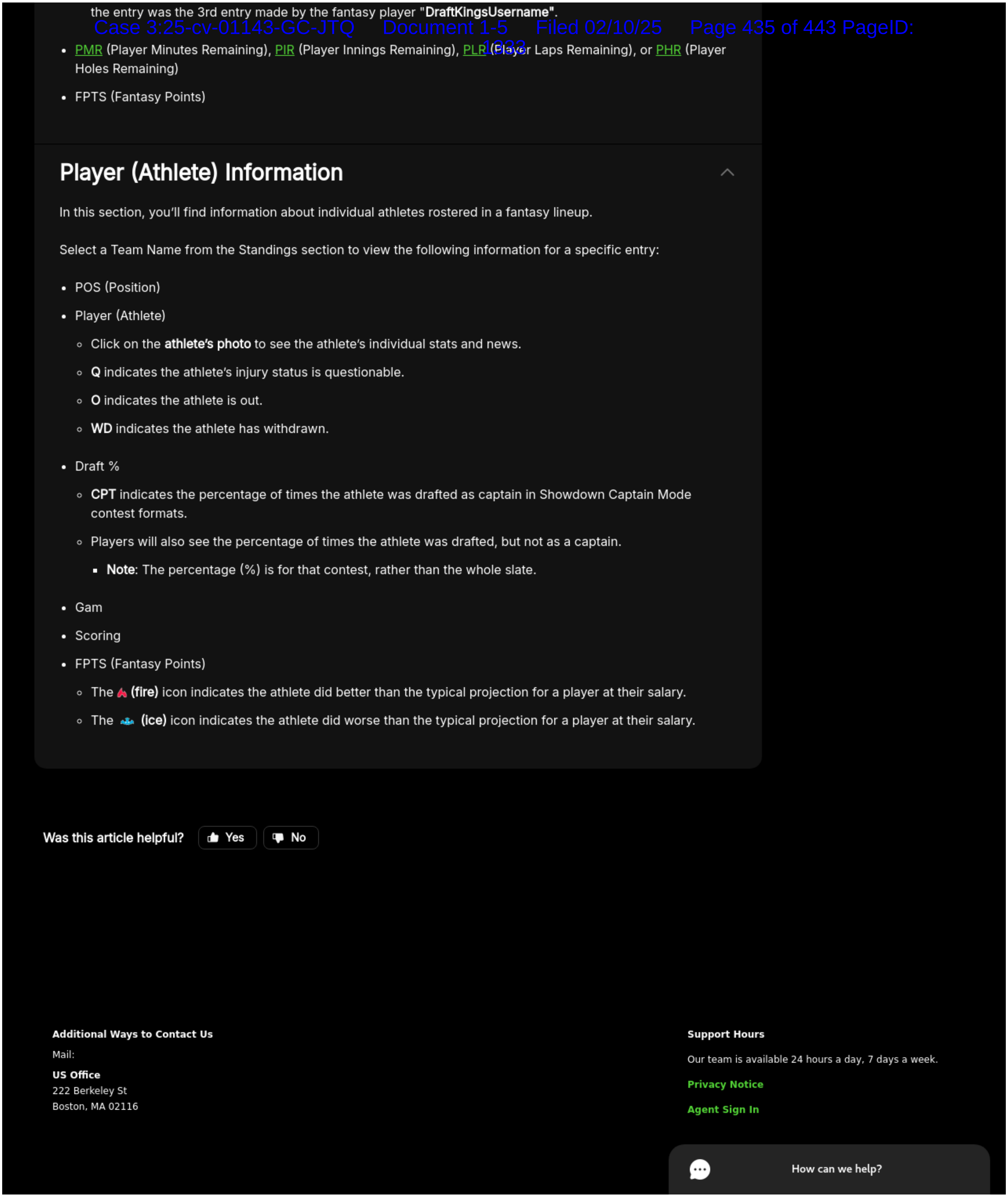
Player (Athlete) Information

In this section, you'll find information about individual athletes rostered in a fantasy lineup.

Select a Team Name from the Standings section to view the following information for a specific entry:

- POS (Position)
- Player (Athlete)
  - Click on the **athlete's photo** to see the athlete's individual stats and news.
  - **Q** indicates the athlete's injury status is questionable.
  - **O** indicates the athlete is out.
  - **WD** indicates the athlete has withdrawn.
- Draft %
  - **CPT** indicates the percentage of times the athlete was drafted as captain in Showdown Captain Mode contest formats.
  - Players will also see the percentage of times the athlete was drafted, but not as a captain.
    - **Note**: The percentage (%) is for that contest, rather than the whole slate.
- Gam
- Scoring
- FPTS (Fantasy Points)
  - The 🔥 **(fire)** icon indicates the athlete did better than the typical projection for a player at their salary.
  - The 🧊 **(ice)** icon indicates the athlete did worse than the typical projection for a player at their salary.

How can we help?



the entry was the 3rd entry made by the fantasy player "**DraftKingsUsername**".

- PMR (Player Minutes Remaining), PIR (Player Innings Remaining), PLR (Player Laps Remaining), or PHR (Player Holes Remaining)
- FPTS (Fantasy Points)

Player (Athlete) Information

In this section, you'll find information about individual athletes rostered in a fantasy lineup.

Select a Team Name from the Standings section to view the following information for a specific entry:

- POS (Position)
- Player (Athlete)
  - Click on the **athlete's photo** to see the athlete's individual stats and news.
  - **Q** indicates the athlete's injury status is questionable.
  - **O** indicates the athlete is out.
  - **WD** indicates the athlete has withdrawn.
- Draft %
  - **CPT** indicates the percentage of times the athlete was drafted as captain in Showdown Captain Mode contest formats.
  - Players will also see the percentage of times the athlete was drafted, but not as a captain.
    - **Note**: The percentage (%) is for that contest, rather than the whole slate.
- Gam
- Scoring
- FPTS (Fantasy Points)
  - The 🔥 **(fire)** icon indicates the athlete did better than the typical projection for a player at their salary.
  - The ❄ **(ice)** icon indicates the athlete did worse than the typical projection for a player at their salary.

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

Exhibit 73

DraftKings Help Center (US) > Fantasy Sports > Contest Rules & Scoring

Can I see my contest standings compared to others in GameCenter? (US)

The GameCenter**'s Standings** section will show the current rank for each entry in the contest, up to 500 entries. For contests that include more than 500 entries, each Team Name will be listed up to the 500th position, and after that only the last place Team Name for each prize tier will be listed.

Important Information

Live contest scoring is strictly for entertainment purposes. Only completed games in the **History** section will display DraftKings' official scoring of the contest, and this validation process doesn't begin until all games within the contest are final.

Contests In Progress

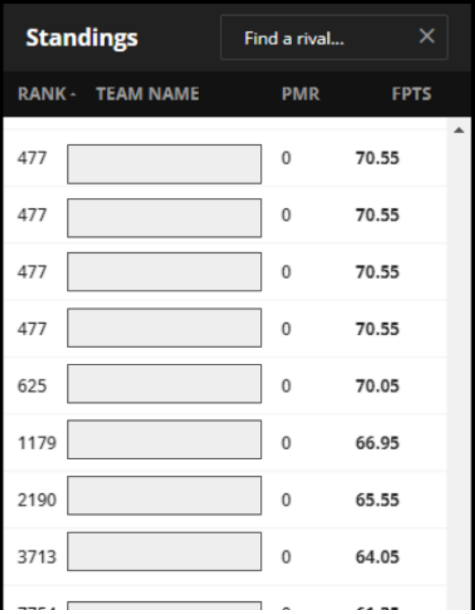
For contests in progress, the standings will be updated regularly, and you'll see changes in rank as points are accumulated in real time.

Contests Marked As Completed

For contests that have been marked as completed, the standings will show each entry's final rank and prize position.

If you're within the top 500 entries at any time, your Team Name name will be displayed in the **Standings** column, or you can use the **Find a rival** search box to locate your own Team Name.

View the visual example below



Standings

Find a rival... ✕

RANK -	TEAM NAME	PMR	FPTS
477		0	70.55
477		0	70.55
477		0	70.55
477		0	70.55
625		0	70.05
1179		0	66.95
2190		0	65.55
3713		0	64.05

JUMP TO SECTION
Fantasy Sports Basics
How to Play- Fantasy Sports
Game Styles & Contest Types
Contest Rules & Scoring
Resolving Fantasy Sports Issues
Tickets
Leagues

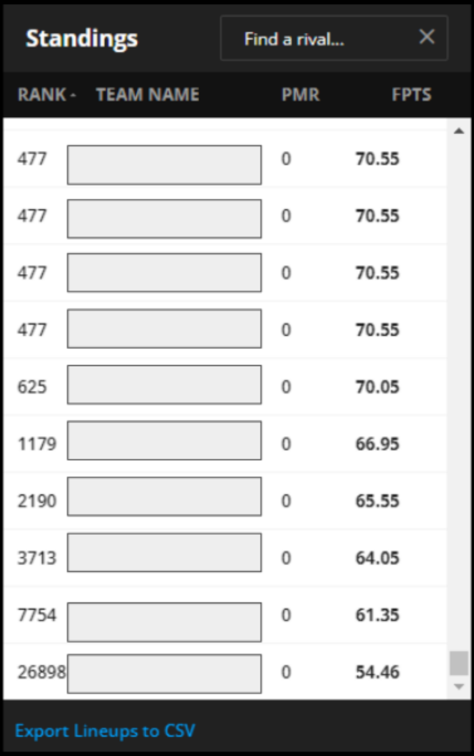
How can we help?

Contests Marked As Completed

For contests that have been marked as completed, the standings will show each entry's final rank and prize position.

If you're within the top 500 entries at any time, your Team Name name will be displayed in the **Standings** column, or you can use the **Find a rival** search box to locate your own Team Name.

View the visual example below



Standings

RANK	TEAM NAME	PMR	FPTS
477		0	70.55
477		0	70.55
477		0	70.55
477		0	70.55
625		0	70.05
1179		0	66.95
2190		0	65.55
3713		0	64.05
7754		0	61.35
26898		0	54.46

- To see a full list of each entry, rank, lineup, FPTS, and more, please visit DraftKings on a desktop device and **Export Lineups to CSV**. Learn more about exporting contest results to CSV.

Was this article helpful? Yes No

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

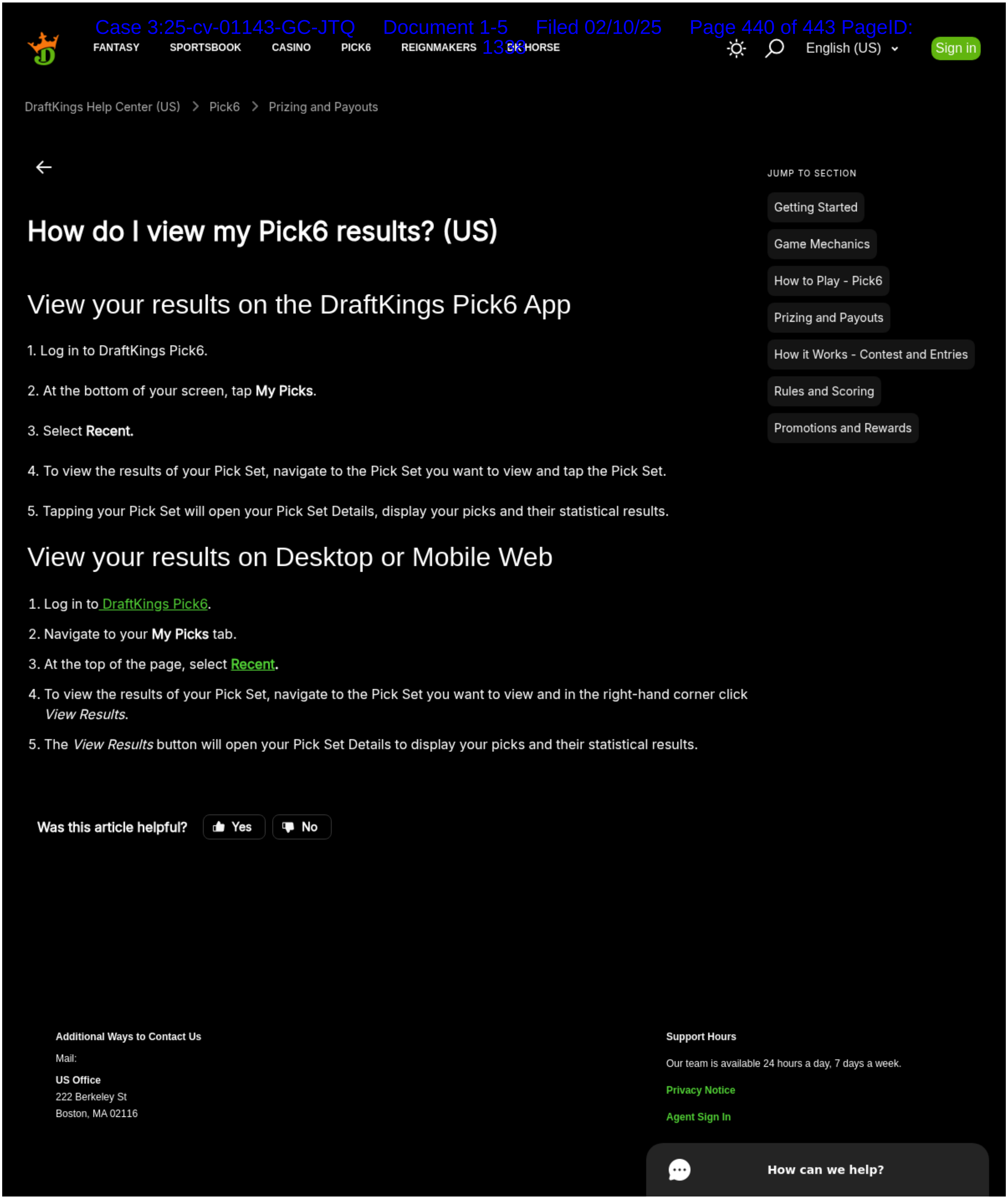
Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

How can we help?

Exhibit 74



How do I view my Pick6 results? (US)

View your results on the DraftKings Pick6 App

1. Log in to DraftKings Pick6.

2. At the bottom of your screen, tap **My Picks**.

3. Select **Recent.**

4. To view the results of your Pick Set, navigate to the Pick Set you want to view and tap the Pick Set.

5. Tapping your Pick Set will open your Pick Set Details, display your picks and their statistical results.

View your results on Desktop or Mobile Web

1. Log in to DraftKings Pick6.
2. Navigate to your **My Picks** tab.
3. At the top of the page, select **Recent**.
4. To view the results of your Pick Set, navigate to the Pick Set you want to view and in the right-hand corner click *View Results*.
5. The *View Results* button will open your Pick Set Details to display your picks and their statistical results.

Additional Ways to Contact Us

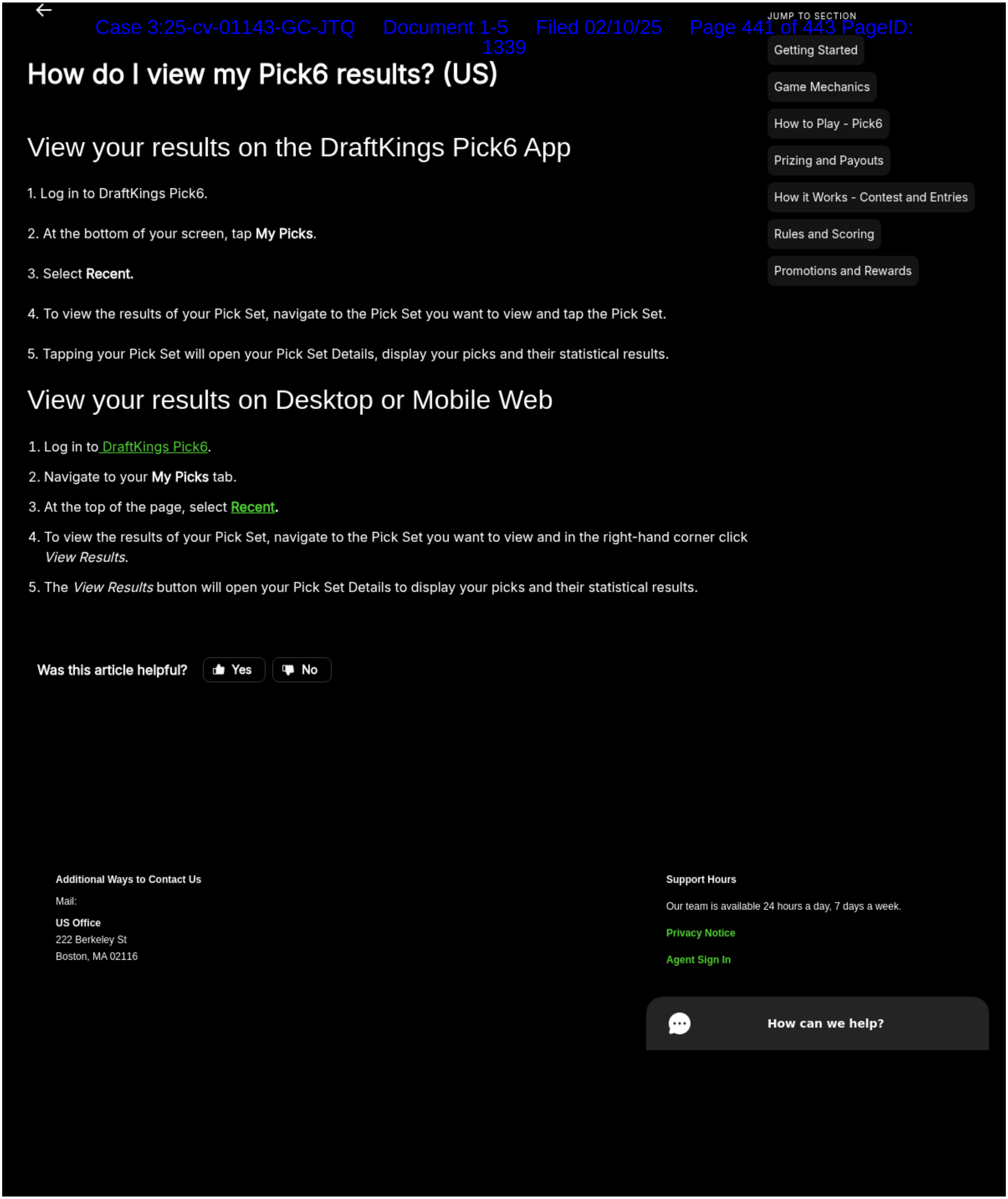
Mail:

US Office
222 Berkeley St
Boston, MA 02116

Support Hours

Our team is available 24 hours a day, 7 days a week.

Document title: How do I view my Pick6 results? (US) – DraftKings Help Center (US)
Capture URL: https://help.draftkings.com/hc/en-us/articles/22657377531539-How-do-I-view-my-Pick6-results-US
Capture timestamp (UTC): Tue, 28 Jan 2025 20:57:03 GMT



How do I view my Pick6 results? (US)

View your results on the DraftKings Pick6 App

1. Log in to DraftKings Pick6.

2. At the bottom of your screen, tap **My Picks**.

3. Select **Recent.**

4. To view the results of your Pick Set, navigate to the Pick Set you want to view and tap the Pick Set.

5. Tapping your Pick Set will open your Pick Set Details, display your picks and their statistical results.

View your results on Desktop or Mobile Web

1. Log in to DraftKings Pick6.
2. Navigate to your **My Picks** tab.
3. At the top of the page, select **Recent.**
4. To view the results of your Pick Set, navigate to the Pick Set you want to view and in the right-hand corner click *View Results*.
5. The *View Results* button will open your Pick Set Details to display your picks and their statistical results.

Additional Ways to Contact Us

Mail:

US Office
222 Berkeley St
Boston, MA 02116

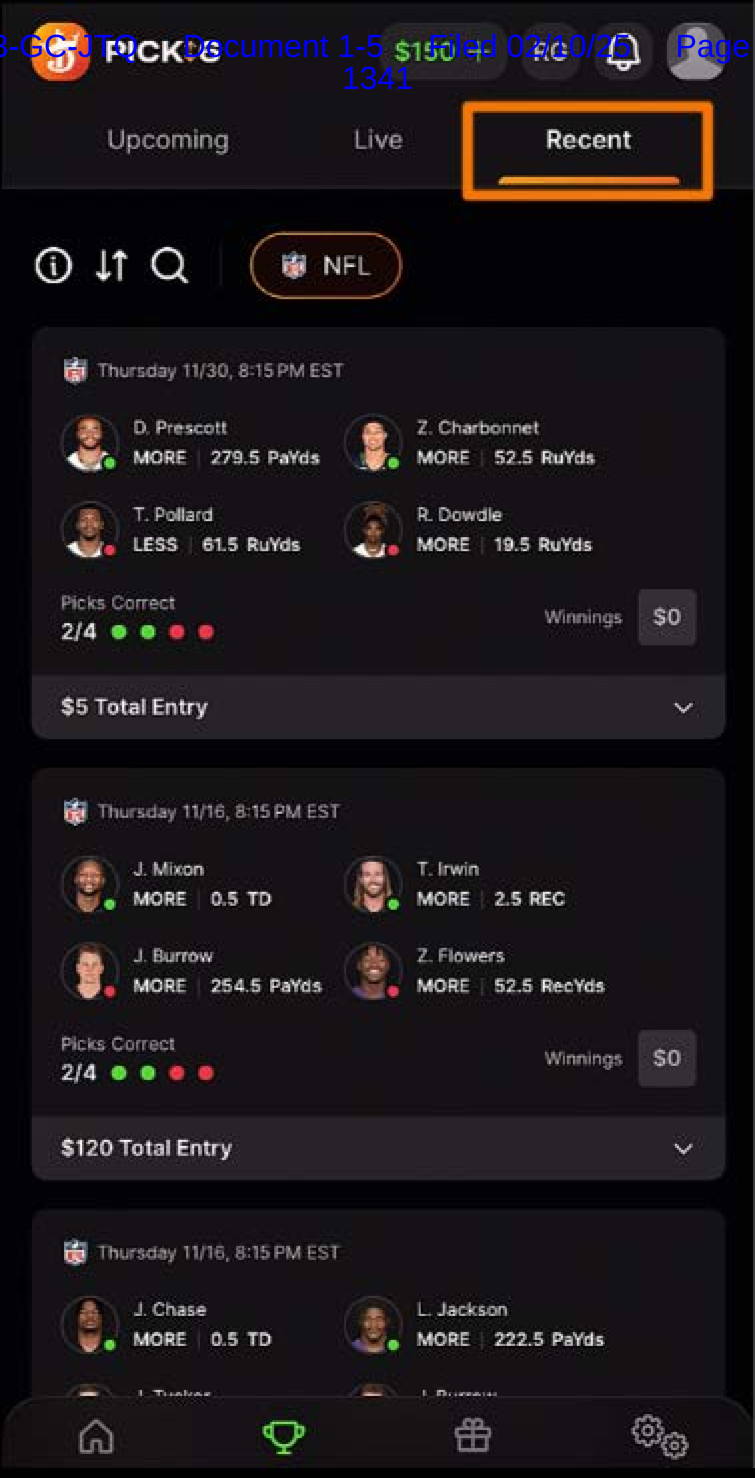
Support Hours

Our team is available 24 hours a day, 7 days a week.

Privacy Notice

Agent Sign In

Exhibit 75



Thursday 11/30, 8:15 PM EST

D. Prescott
MORE | 279.5 PaYds

Z. Charbonnet
MORE | 52.5 RuYds

T. Pollard
LESS | 61.5 RuYds

R. Dowdle
MORE | 19.5 RuYds

Picks Correct
2/4 🟢 🟢 🔴 🔴

Winnings $0

$5 Total Entry

Thursday 11/16, 8:15 PM EST

J. Mixon
MORE | 0.5 TD

T. Irwin
MORE | 2.5 REC

J. Burrow
MORE | 254.5 PaYds

Z. Flowers
MORE | 52.5 RecYds

Picks Correct
2/4 🟢 🟢 🔴 🔴

Winnings $0

$120 Total Entry

Thursday 11/16, 8:15 PM EST

J. Chase
MORE | 0.5 TD

L. Jackson
MORE | 222.5 PaYds

J. Tucker

J. Burrow